Modeling and Simulation of Circumstellar Disks with the Next Generation of Hydrodynamic Solvers

Modeling and Simulation of Circumstellar Disks with the Next Generation of Hydrodynamic Solvers

A dissertation presented

by

Diego José Muñoz Anguita

to

The Department of Astronomy

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in the subject of

Astronomy & Astrophysics

Harvard University

Cambridge, Massachusetts

August 2013

Dissertation Advisor: Professor Lars Eric Hernquist Diego José Muñoz Anguita

Modeling and Simulation of Circumstellar Disks with the Next Generation of Hydrodynamic Solvers

Abstract

This thesis is a computational study of circumstellar gas disks, with a special focus on modeling techniques and on numerical methods not only as scientific tools but also as a target of study. In particular, in-depth discussions are included on the main numerical strategy used, namely the moving-mesh method for astrophysical hydrodynamics. In this work, the moving-mesh approach is used to simulate circumstellar disks for the first time.

The structure of the thesis follows a natural progression that begins by discussing the role of computational methods in modern astrophysics, followed by a description of the moving-mesh method as a general solver for gas dynamical problems, and concluding with detailed modeling of circumstellar disks in two and three dimensions, both in isolation and in pairs.

The thesis structure consists of two parts. Part I –second and third chapters– focuses on moving-mesh hydrodynamics and Voronoi meshes in general, deriving the discretized equations of the method from first principles and describing the time-stepping technique in detail. This section also includes original work on numerical methods to include diffusion terms to the equations of hydrodynamics, such as physical viscosity.

In Part II of the thesis –fourth, fifth and sixth chapters– the attention is turned to circumstellar disks. In the fourth chapter, two-dimensional disk simulations are

carried out as a benchmarking stage, before more complex, three-dimensional models can be pursued. Novel techniques for creating stable, three-dimensional models of self-gravitating disks with finite radius are discussed in the fifth chapter. In this model, the Voronoi discretization of the computational domain allows for a smooth transition between the mesh that discretizes the disk and the mesh that discretized the background space. Details are provided on how stationary models can be created *a priori* without the need for relaxation procedures as done in previous work.

Finally, the sixth chapter includes a set of simulations that, owing to their complexity, require a scheme that combines the features of the method discussed in preceding chapter. Specifically, such a scheme must be capable of treating self-gravitating systems that (1) lack an obvious symmetry, (2) include regions of high-Mach number flow, (3) have a large dynamical range in density and (4) need an adaptive mesh resolution to adequately capture strongly compressed/shocked regions and potentially fragmentation. To this end, a suite of novel simulations of disk-disk interaction is carried out, to conduct an initial study of the tidal effects that massive disks have on the evolution of their host stars' orbits.

Contents

Acknowledgments

I want to thank my family, Mauricio, Paulina, Rosario, Paulina and Gonzalo, for their constant support, encouragement and love over my entire life, especially during the very long years we have been apart. I want to thank Maria for her generous and unconditional help. I am thankful to all of them for making our home in Chile truly feel like home every single time I visited, as if I had never left.

I am grateful to have had Lars Hernquist as an advisor. I greatly appreciate the freedom he has allowed me and which has helped me become an independent researcher. Group meetings with the Hernquist group will be missed. Above all, I thank Lars for taking me as a student when I was already in my fourth year, giving me an opportunity to give astronomy a second chance.

I am very thankful of other researchers and postdocs who have helped me with their insightful suggestions and advice. I would like to thank Kaitlin Kratter, for her constant disposition to talk about research and her invaluable help with my thesis work; Mark Vogelsberger for his availability and invaluable technical expertise; Ruth Murray-Clay for her great physical insight and especially for support during tough periods; Jim Moran, who helped me realize I wanted to be a theorist. I am thankful to Matt Holman and Dimitar Sasselov for great conversations about science. I also want to give special thanks to Volker Springel, without whose enormous contribution to astrophysics this thesis would have not been possible.

I am thankful to all the wonderful friends I made in graduate school. Thanks to Jack , Robert H, Joey, Andy, Robert M., Gurtina, Claude-André, Sarah B., Sarah R.,

Jaime, César, Andrés, Lucho, José Miguel, Nick, Greg, Zach, Paul, Wen-Fai, Dylan and especially my former officemate Chris, who is partially responsible for making my stay in grad school longer that it should have been.

I want to thank my childhood friends in Chile and overseas, Juan Pablo, Ismael, Cristián, Cristóbal, Rodrigo and many others for their unwavering friendship despite the distance between us and my irredeemable inability to keep in touch.

And finally, I want to thank Laura for her support and company, for caring, for being the way she is and, most of all, for understanding.

A Mi Familia

1

Introduction

This introductory chapter of my doctoral thesis aims to introduce the reader to the role of numerical modeling and simulation in astrophysics in general, and circumstellar disks in particular.

Circumstellar disks are the gas and dust structures in which planets form. Understanding the physical processes that give origin and dictate the evolution of circumstellar disks is essential to understanding the initial conditions that can give rise to planetary systems like our own Solar System, as well as the incredibly diverse range of planetary configurations now known to exist throughout the Galaxy.

The basic concept of planet formation as a by-product of star formation, in which planets form out of a flattened rotating cloud of gas and dust, can be traced back as far as the works of Emanuel Swedenborg and Immanuel Kant (Swedenborg, 1734; Kant, 1755). This original "nebular hypothesis" was revised and reformulated several times for over two centuries, before Victor Safronov (Safronov, 1972) developed the modern

framework from which most modern theories of planet formation are derived. Although the precise process of planet formation is still an incomplete theory, there is unanimous consensus that planets form while embedded in circumstellar disks, and that this process must take less than 10 million years, which is the measured lifetime of primordial gas disks around young stars (e.g. Haisch et al., 2001).

The observational evidence has conclusively shown that planets form in our Galaxy outside our own Solar System[1] . Similarly, we know circumstellar disks exist around young stars. We know that these disks are common in star-forming regions and that they rapidly dissipate as their host stars enter the main sequence. What happens in between the gas disk phase and the planetary system end phase is still under debate. The complex process that dictates planet formation can only be studied in detail through direct numerical simulation. Computational research on this front will be especially important during the time in which we have no direct evidence of a planet caught in formation.

Idealized models of planet formation and planet-disk interaction have been studied numerically for two decades now. Despite the enormous progress that can be achieved by the quantitative approach of computational modeling, these studies are often still based on very simplified models of the real systems. One simplification that might hide important features of planet formation is the neglecting of multiplicity in star formation. Stars often form in binaries, and sometimes in hierarchical multiples, creating a more tumultuous and dynamically complex physical environment than isolated circumstellar disk models can capture. A self-consistent, deterministic and general theory of planet

[1]As of May 22nd, 888 (confirmed) exoplanets have been identified: http://exoplanet.eu/catalog/

formation will have to eventually explain the role that stellar multiplicity plays in the evolution of planet-forming gas disks. Being essential for model building and theory-testing, numerical simulation must move in this direction, making models as realistic as theory and computational resources allow. This effort must incorporate more physics in our simulations at the fundamental level of equation-solving, as well as with the inclusion of peripheral models or prescriptions. Numerical techniques *themselves* will play a major role in improving the power, quality and reliability of our simulation output. This thesis explores new techniques for circumstellar disk simulations in complex environments.

1.1 Astrophysics in the Age of Supercomputers

The scientific method has witnessed a revolution for the past 50 years: the rise of computer simulation. This method of research has changed the way scientist carry out experiments, becoming indispensable in physics, engineering, chemistry, biology and other disciplines. Astrophysics is not the exception to the "computer simulation revolution", and in some aspects, it has become a leading field in the search for bigger, better and faster methods and numerical experiments.

The birth of computational methods – in particular, of computational physics – goes back to Hungarian-born American mathematician and Manhattan Project scientist John von Neumann, who foresaw the importance of programmable computers when the first general-purpose machine (the Electronic Numerical Integrator And Computer, ENIAC) first came online in 1947. ENIAC was originally intended for ballistic research, but von Neumann became so involved with the development of this machine that its

first computations were on models for the hydrogen bomb, in which von Neumann had been working at Los Alamos with physicist Edward Teller (a.k.a. "the father of the hydrogen bomb") [2] . Under Teller's encouragement, von Neumann recruited Manhattan Project members Stanislaw Ulam and Nicholas Metropolis to study problems of nuclear reactions. One of the outcomes of this collaboration at Los Alamos was the internal report titled "Statistical Methods in Neutron Diffusion", by Ulam and von Neumann in addition to Robert D. Richtmyer, which records the first use of the Monte Carlo method (Ulam et al., 1947). In 1950, von Neumann and Richtmyer would introduce the concept of "artificial viscosity" for shock capture in finite-difference methods for the Euler equations (von Neumann & Richtmyer, 1950). This technique would prove indispensable in keeping shocks and discontinuities reasonably well-resolved, while avoiding numerical instabilities and non-monotonic oscillations that arise in strictly conservative schemes.

John von Neumann's first developments for numerically solving the Euler equations mark the beginning of the field of computational fluid dynamics (CFD), a discipline that is essential nowadays for computational astrophysics and the main focus of this thesis.

The first use of computers in astrophysics *per se* did not come from gas dynamics (that would have to wait until the late 1970s and early 1980s) but from the numerical integration of the N-body problem. Unsurprisingly, the same phenomena that put the "physics" in astrophysics in the 16th century –namely, the motion of celestial bodies under gravitational forces– introduced computers to astrophysics. In the early 1960s, Sebastian von Hoerner (von Hoerner, 1960, 1963) and Sverre Aarseth (Aarseth, 1963) carried out direct N-body integrations of globular clusters (with about 100 particles).

[2]Teller and von Neumann were both veterans of the Manhattan Project.

Around the same time, Cohen & Hubbard (1965) carried out the first long term integration of the five outer planets of the Solar System using the Naval Ordnance Research Calculator (the IBM NORC, the most powerful computer at the time of its building in 1954), in a calculation that took ~ 80 CPU hours and that ran the Solar System backward in time for 120000 years. Cohen et al. (1973) would extend the integration time to 10000000 years. Before the work of Cohen and Hubbard, studies on the long-term behavior of the Solar System was carried out using perturbation theory, in most cases the low-order secular theory (e.g. Brouwer & Clemence, 1961; Murray & Dermott, 2000). During the 1970s, the N-body calculations started focusing on large scale structure and galaxy formation. The first cosmological simulations were part of the famous Press & Schechter (1974) paper on hierarchical clustering and the small-number simulations of Haggerty & Janin (1974) (see Bertschinger, 1998 and references therein). The filamentary nature of the large-scale structure of the Universe was first seen in simulations by Frenk et al. (1983) (with models that included only 1000 particles). The late 1980s and early 1990s saw a new breakthrough in N-body calculations afters the introduction of tree codes to calculate gravitational forces (Barnes & Hut, 1986; Hernquist, 1987), but these approximations would not reach their full power until massively parallel machines became available in the late nineties for scientific computing. This computational revolution meant a enormous increase in memory and computing power that allowed for a quantum leap in the size of simulations (previously limited to 10^6 particles for single-processor machines). New parallel gravity solvers (e.g., Dubinski, 1996; Makino et al., 1997) were crucial for developments of codes like GADGET-2 (Springel, 2005), which was used to cross the 10 billion particle mark for the first time with the groundbreaking cosmological *Millenium Simulation* (Springel et al.,

2005b).

While N-body method were being developed between the 1960s and the 1980s, CFD was reaching its maturity thanks to the contributions of people like Sergei Godunov, Peter Lax, Burton Wendroff and Bram van Leer among others, who cemented the bases for the finite-volume method for hyperbolic conservation laws and the incorporation of Riemann solvers (see Toro, 2009) to update hydrodynamic quantities in what is now universally known as the Godunov method. However, before these grid-based method became commonplace in computational astrophysics, a different scheme made its way into the numerical study of gas dynamical problems in astrophysics: smoothed particle hydrodynamics (SPH).

SPH was developed by Gingold & Monaghan (1977) and Lucy (1977) for simulations of three-dimensional rotating stars. Since then, SPH has been a major resource to many astrophysicists carrying out numerical simulations from cosmological to planetary scales. After SPH was introduced, grid methods found their way into astrophysics after the introduction of the piece-wise parabolic method (PPM) by Colella & Woodward (1984) paving the way for grid-based schemes for astrophysics in subsequent years (e.g. Evans & Hawley, 1988; Stone & Norman, 1992; Klein et al., 1994; Fryxell et al., 2000; Stone et al., 2008; Mignone et al., 2012). However, SPH has remained a popular and important method for astrophysical problems.

Some of the first global simulations of circumstellar disks were done using SPH. (e.g. Artymowicz et al., 1991; Artymowicz & Lubow, 1994; Murray, 1996). The obvious advantage of SPH for simulating boundary-free systems made it the tool of choice for complicated systems. Before the work of Armitage (1998), most grid-based simulations of

disks had concentrated on the shearing sheet/box approximation (e.g., Hawley & Balbus, 1991; Hawley et al., 1995; Stone et al., 1996; Gammie, 2001). Both SPH and grid-based codes remain popular among simulationists for problems with different characteristics. Just like with the collisionless N-body problem, the increment in computer power added to parallel programming libraries such as the message passing interface (MPI) have enabled an enormous increase in resolution over the past decade, with examples such as the disk instability simulations of Meru & Bate (2011) with ~ 20 million particles and the global MHD simulations for Sorathia et al. (2012) with ~ 120 million cells.

1.2 What is an Astrophysical Simulation?

The fast growth of simulation has prompted a debate among philosophers of science in recent years in regards to what simulations are and where they lie in the methodology map. In practice, simulations have elements of pure theory and elements of pure experiment. This fact is used by some to argue there is "nothing new" about simulations since they recycle the same old scientific methods but just using digital or "in silico" laboratories. Conversely, others argue that the very hybrid nature of computational research is what demands defining a new category for the way simulations generate new knowledge (what philosophers like to call "the epistemology of simulation).

Regardless of what the "epistemic category" of simulation research is, it is empirically true that computer simulations fill a gap where pure theory is not able to explain the observed phenomena, and where experiment is not able to explore or reach the physical conditions under study. This is especially true for the broad field

of astrophysics[3] . Among empirical sciences, astrophysics suffers from an unavoidable scarcity of data, as it is inherently observational rather than experimental. Photons observed through telescopes are the only available data on real systems. Thus, numerical simulation plays an essential role in astrophysics as a form of "laboratory" in which experiments are controllable and repeatable.

The fact that astronomers cannot setup controlled experiments the way physicists do puts Astronomy in a special place within the natural sciences, one which might not satisfy all the "requirements" of a traditional "hard" science[4] , and one which requires sophisticated statistical techniques to sanction physical models for objects. The scarcity of data (in the traditional, controlled-experiment sense) is the main motivator for computational research in astrophysics. As such, it is the goal of the "simulationist" is to use computer simulations as useful and valid *generators of knowledge* next to traditional observational and theoretical astrophysics.

The term "simulation" is often used loosely, and sometimes researchers will not agree on whether to call their computational models and results "numerical experiments"

[3]Some would argue here for the use of "Astronomy" as the general field, and that *Astronomy* is the overall discipline encompassing astrophysics and other "astro-sciences". Indeed, that is the definition held by the Merriam-Webster dictionary:

> **astrophysics**: a branch of astronomy dealing especially with the behavior, physical properties, and dynamic processes of celestial objects and phenomena

However, this is a definition that I personally dislike, essentially for historical and etymological reasons, given that -nomy comes from nomos ($\nu\acute{o}\mu o\varsigma$), meaning "law" or "culture", and thus Astronomy, in the proper ancient Greece context, is the compilation of nomenclature or the categorization of celestial bodies, and not the study of the physical process that explain their behavior and/or existence.

Perhaps a more precise terminology should include "logos" ($\lambda\acute{o}\gamma o\varsigma$), which was used by Heraclitus (ca. 535-475 BC) as a principle of order and knowledge, but I seriously doubt any self-respecting scientist would choose to call her field of study "Astrology".

[4]Under some definitions (e.g. Lemons, 1996, pp 99; Rose, 1997, § 1; Diamond, 1987), a "hard" (rigorous) scientific discipline is one that: (1) produces testable predictions; (2) performs physical controlled experiments; (3) relies on quantifiable data and mathematical models; (4) possesses a high degree of accuracy and objectivity; (5) and generally applies a purer form of the scientific method.

or "numerical simulations". Semantics is a trap that scientist will not fall into often, leaving such metaphysical exercises to philosophers. However, I will allow myself the indulgence of discussing the meaning of numerical simulations because computers pervade every single aspect of modern astronomy and astrophysics, however, not every use of a computer qualifies as a simulation (of a physical object of astronomical interest).

A common-place definition of simulation can be the following:

Definition 1. *Simulation*
A stand-in or mimic of a real-world system.

This is a definition that some philosophers of science adhere to. However, it is a very general definition. For starters, it applies to any type of stand-in, or replacement, regardless of it being carried out by a digital computer. In fact, this loose definition is sometimes used by philosophers of science (like Eric Winsberg) who support the idea that numerical research is an entirely new methodology of science. This definition allows some experiments to be catalogued as analog simulations: laboratory experiments are idealized, controlled settings that try to isolate some phenomenon that takes place in nature, but they are not the *actual* target of study after all. Then, according to these supporters, the fact that we can now simulate digitally instead of analogically is itself a technological and scientific revolution.

Another very common definition, and perhaps a scientist favorite, is

Definition 2. *Simulation*
The use of a computer to solve an equation that we cannot solve analytically.

which is the one favored by philosophers of science who dismiss the idea that

computer simulation presents an entirely new scientific method like Roman Frigg and Julian Reiss (see Frigg & Reiss, 2009). However, this definition puts computer simulation at the same level as any numerical computation like Gaussian quadrature to evaluate integrals, the Newton-Raphson scheme to solve algebraic equations, the Runge-Kutta method to solve differential equations, etc (note that all the examples I gave predate, some of them by centuries, the existence of digital computers). As Humphreys (2009) said (see also Humphreys, 2004), computer simulation must be distinguished from the numerical methods it uses and from numerical methods in general.

Let us try a combination of the definitions by Humphreys (2009), Winsberg (2010) and Roache (2012) plus a few additions of my own:

Definition 3. *Computer Simulation*
The entire process of constructing, using and justifying a *dynamical* model that involves analytically intractable mathematics (often involving partial differential equations), as well as ad-hoc and peripheral models, and that is evolved/integrated *in time* by using numerical techniques programmed onto a digital computer.

Among other things, I have added the line "involving partial differential equations" because, although this is not a strict requirement of the definition, the complexity leap of going from ordinary differential equations (ODEs) to partial differential equations (PDEs) is essential in understanding that simulations use imperfect methods and imperfect approximations, and the simulation research is always trading off "rigor for expediency" (Winsberg, 2010). More pragmatically, the explicit inclusion of PDEs narrows down our concept of simulation to the approximate solution of continuous physical systems, in particular problems of compressible fluid flow.

It is my opinion that the two concepts of "numerical simulation" and "numerical experiment", albeit not synonymous, overlap greatly. However, they are completely distinguishable (as I have tried to express above) from the concepts of "numerical integration" or "numerical calculation." To illustrate this point, I reproduce a conversation that took place in my presence two years ago at the Harvard-Smithsonian Center for Astrophysics:

> **Chris Hayward**: I must say that the more I learn about simulations, the less I trust them!
>
> **Matthew Holman**: Really?! In my case, the more I learn about simulations, the *more* I trust them.
>
> **Chris Hayward**: Fair enough. I guess that *your* simulations are more trustable then!

Computer simulations are neither truth nor fiction, they are model results. And depending on the degree of approximation (either physically motivated or pragmatically motivated) the results will have different degrees of reliability and connection to the real world. This multi-stage, multi-component process can be represented in a simplified diagram as the one shown in Figure 1.1 (see also Figure 2.1 in Winsberg, 2010).

Dr. Holman and Dr. Hayward are indeed talking about different types of simulations. Dr. Holman's research in Solar System dynamics and planetary systems make use of high-accuracy integrators for N-body systems with exact (Newtonian) gravity-force computations. In this case, the physical system is written *directly from* the theory as a coupled system of ordinary differential equations (ODEs). The approximation

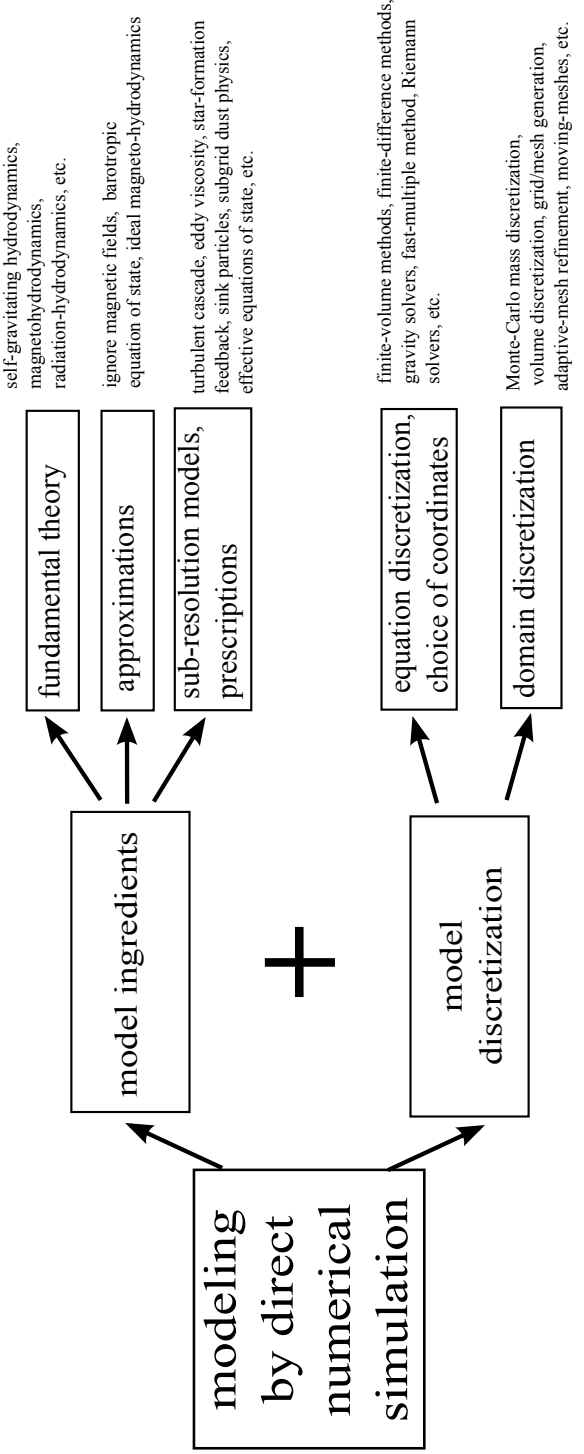


Figure 1.1.— Diagram exemplifying the components of the simulation methodology.

enters when discretizing the time variable in the ODE in order to evolve the system forward in time. The time-evolution methods used are of extraordinary accuracy and are physically motivated. Practical approximations to these calculations are, for example: assuming planets and Solar System bodies are point masses, ignoring for example General Relativity or the effects of millions of minor bodies. All these approximations are *very* good approximations, and consequently, the model is very close *conceptually* to the underlying theoretical framework.

Dr. Haywayrd's research is about the evolution of galaxies. Besides gravity, there are additional physical components that make this system differ significantly from the Solar System, and thus, the computational techniques to treat it will differ as well. For starters, the underlying theory is self-gravitating hydrodynamics, which now deals with coupled PDEs instead of ODEs, and which necessarily include a discretization of space. In addition to resolution limitation, galaxy formation models often include observationally-motivated, sub-resolution models for phenomena known to occur on small spatial scales.

We can conclude that, within computational astrophysics, there is a large variation between models conceptually close to the underlying theory and models that have a large additional content that is *motivated* by the real world and therefore necessary to accomplish an adequate mimicking of the target of study.

The use of computational techniques to perform calculations is not the same as a full-blown numerical simulation. Quoting Winsberg again, there are at least two characteristics that distinguish "true simulations" from brute-force numerical computations:

> (1) Successful simulation studies do more than compute numbers; they make use of a variety of techniques to draw inferences from these numbers. (2) Simulations make creative use of calculational techniques that can only be motivated extra-mathematically and extra-theoretically. As such, unlike the results of simple computations[...], the results of simulations *are not automatically reliable* (Winsberg, 2010, pp 32, emphasis added here)

This explains nicely the differences between the different kinds numerical studies presented in the Holman-Hayward example. A competitor of Holman might disagree with Holman's results based on the most fundamental of the approximations used (say, ignoring GR or the presence of the most massive Kuiper Belt objects), but would rarely challenge the "Treatment" step for integrating the equations of motion. On the other hand, a competitor of Hayward might agree on all the subgrid physics models used by him, but might distrust the reliability of his results solely because Hayward used SPH instead of a grid-bases method, disagreeing with the *treatment* of the discretized equations.

One can go further in this comparison, by comparing algorithmic structures, the accuracy of the integrators, time-marching schemes, etc. The more populated the diagram in Figure 1.1 is, the farther from pure theory our methods lie. Consequently, it is only experimenting, validating, verifying and benchmarking that we can build credentials for a simulation method.

In computer simulations, the code itself can be the object of study. This may come to a surprise to some, but the intricate and highly degenerate process of code validation and verification necessarily implies testing codes against each other, in different regimes,

with different initial conditions, with different sub-grid prescriptions,etc. Thus, computer simulation research often entails *experimenting* with the algorithmic tools we have originally built to study a different target system. This abstraction from the natural world has created some doubts amongst philosophers (and even scientists) and has spun some debate over the last decade on whether or not a new philosophy of science (i.e. a new epistemology) is required for computer simulations.

1.3 Circumstellar Disks on a Moving Mesh: Flexible Approaches for Complex Geometries

Now, after discussing the context of my research – computer simulations in astrophysics – I will explain in further detail the benefit and/or necessity of exploring novel numerical techniques for the analysis of systems of astrophysical interest. Novel techniques present an opportunity to remove degeneracies in the systematic errors intrinsic to the computational methodology. New discretization and integration techniques allow us to study systems that have *aready been studied* with other techniques, as well as systems that are either *too difficult or impossible to be studied* with preexisting approaches.

The experimental nature of simulation requires the reexamination of the same problems with different tools. One very recent example of this approach producing new knowledge about simulation techniques is the collective "Moving-Mesh Cosmology" effort being underway jointly at the Heidelberg Institute for Theoretical Studies and the Harvard-Smithsonian Center for Astrophysics (see Vogelsberger et al., 2012; Kereš et al., 2012; Sijacki et al., 2012; Torrey et al., 2012; Nelson et al., 2013). These studies found

that the formation of flattened disks at the center of dark matter haloes (i.e. a "galaxy") in *ab-initio* cosmological simulations could be strictly explained by the hydrodynamical solver being used – a moving-mesh technique versus an SPH technique – instead of by the subresolution and ad-hoc physics included in the models such as thermal feedback (see, for example, Governato et al., 2004; Guedes et al., 2011).

The "discretization" box in Figure 1.1 lies at the core of the research I present in this dissertation. In the sections below, I explain why new methods for hydrodynamics should be applied to circumstellar disk simulations, and how the moving-mesh approach introduced by Springel (2010a) presents a unique opportunity to explore regions of the parameter space that were not easily accessible to other existing numerical techniques.

1.3.1 Why Moving Mesh Methods?

Although the formation of a flattened, rotating structure around a young stellar object is a natural consequence of angular momentum conservation during gravitational collapse of a dense core, direct *resolved* imaging of such objects was not possible until the late 1980 and early 1990s. (Beckwith et al., 1986; O'dell et al., 1993; McCaughrean & O'dell, 1996; Mundy et al., 1996; Wilner et al., 1996). Before that, the existence of dusty disks was inferred from the near-infrared excess in the T-Tauri star spectra (e.g. Mendoza V., 1966). The evident challenges in spatially resolving these objects has begun to be overcome in the last decade, particularly thanks to (sub-)millimeter interferometry, which has allowed the mapping of the fainter sources, and the resolution of disk sizes and inner cavities (e.g. Hughes et al., 2008, 2009) .

Up until recently, disk images have not had sufficient resolution to show fine

structure (at sub-AU scales). As a consequence, the observed surface densities have been successfully modeled in simple parametric form (e.g. Andrews et al., 2009, 2010b). The level of complexity of these analytic models has varied little since the early theory of accretion disks (e.g. Shakura & Sunyaev, 1973; Pringle, 1981), nevertheless providing enough physical insight into the structure of protoplanetary disk. However, recent observations in mid-IR scattered light (Muto et al., 2012; Grady et al., 2013) and CO/HCO^+ sub-millimeter emission (Casassus et al., 2013) indicate that the finer structure of circumstellar disks can be quite rich and deviate significantly from axisymmetry. The limitation of parametric models can be overcome by direct numerical simulation, which enables us to capture the gas dynamical evolution of these systems self-consistently and study complex configurations in detail.

Another major departure from the ideal axisymmetric disk model is that caused by planet and stellar multiplicity. One of the most striking features of the *Kepler* catalogs (e.g. Batalha et al., 2013) is the widespread presence of multi-planet systems. Multi-planet systems add a significant degree of complexity to some of the physical processes believed to be relevant for the formation and survival of planets, namely planet migration and planet growth through the core and gas accretion phases.

Planet formation is not only plagued with the effects planet multiplicity, but it is also likely to be affected to some extent by the environmental *stellar* multiplicity. Multiplicity rates of pre-main sequence stars are known to be higher than those of main sequence stars (Mathieu, 1994; Kraus et al., 2011), and this must have some influence on the primordial circumstellar disks (e.g. Artymowicz & Lubow, 1994) and the subsequent dynamical evolution of planets (e.g. Adams et al., 2006; Craig & Krumholz, 2013). Recent observations of young multiples in Taurus with both *Spitzer* observations

(e.g. Kraus et al., 2012) and the Submillimeter Array (SMA) (e.g. Andrews et al., 2010a; Harris et al., 2012) reveal increasingly diverse multiples and binaries bearing circumstellar and circumbinary disks, evidencing that the idealized system consisting of one star and one disk might not be representative of the Galactic-wide planet formation process. Despite these increasingly complex configurations, hydrodynamical simulations of planet-forming systems often focus on models of gas disks orbiting one isolated, stationary star.

Circumstellar disks within star-forming environments (including, among other conditions, stellar multiplicity and stellar bulk motion) can be extremely challenging for some of the known numerical techniques. One of such challenges is how the geometry of the system determines which set of coordinates is most appropriate for the formulation of the hydrodynamic equations and their subsequent discretization. It is well known that the performance of Eulerian hydrodynamical codes depends significantly on the geometry of the mesh aligning with the direction of bulk flow [5] . Truncation error analysis allows for the derivation of "model" or "modified" equations (see LeVeque, 2002, §8 and references therein) of which the discrete versions of the equations of motion are exact solutions. These modified equations show that space and time discretization introduces high-order diffusive and dispersive differential operators, the most common of which (the result of first-order upwind schemes LeVeque, 2002; Toro, 2009) is the so-called "false diffusion" (or "numerical viscosity" or "advection error"). This diffusion depends on the fluid velocity and the grid spacing and it is also a function of the angle between the flow direction and the coordinate axis used to discretize the domain (de

[5]One notable exception is that of higher-order finite-difference schemes such as the PENCIL code (sixth-order accurate in space), the performance of which is essentially independent of the geometry of the mesh (Lyra et al., 2008)

Vahl Davis & Mallinson, 1972; Patankar, 1980).

This property of Eulerian codes becomes especially relevant in astrophysical fluid dynamics, where the evolution of the energy equations is often important. In particular, in cosmological simulations with advection-dominated flows (e.g. in extremely supersonic motion near the cosmological density peaks) in which the kinetic energy density is much larger than the internal/thermal energy density, small fractional errors in the velocity can translate into large fractional error in temperature, eventually distorting the thermodynamic evolution of the gas. This has been referred to the "high Mach number problem" (Ryu et al., 1993; Bryan et al., 1995; Feng et al., 2004; Trac & Pen, 2004). The grid-dependent diffusion of Eulerian codes explains why a cylindrical-coordinate discretization of the domain is the preferred choice to study accretion and protoplanetary disks, since the direction of the flow is almost purely azimuthal The high-Mach-number problem can be alleviated in cylindrical coordinates by the use of the FARGO[6] technique (Masset, 2000), which removes the Keplerian speed at a given radius at the moment of numerically solving the Euler equations, thus effectively using a non-inertial moving frame in which the velocity changes are small. However, thin accretion disks are not only highly supersonic in the azimuthal direction (the orbital Mach number is $\mathcal{M} \sim 1/h \sim 50$ for aspect ratios of 0.02), but they can also have supersonic bulk speeds. For young stellar associations, pre-main sequence stars can move at relative speeds of 1-3 km s^{-1}, which is roughly ten times or more the sound speed of molecular gas at a temperature of 10 K.

[6]The most obvious benefit of implementing FARGO (Fast Advection in Rotating Gaseous Objects) is the great increase in the allowed time-step size, but the underlying benefit is reducing the numerical diffusion by ignoring the bulk velocity of the flow and instead solving for the deviations from it.

Evidently, if there is a strong deviation from a point mass potential (e.g. owing to the presence of a large mass ratio binary), or if this point mass is allowed to move across the computational domain, the flow is no longer one-directional in this coordinate system, and the choice of a cylindrical geometry is no longer the most obvious one. In cases where there is no obvious symmetry that can be exploited through a suitable choice of a coordinate frame, numerical studies commonly use either adaptive mesh refinement (AMR; Berger & Oliger, 1984; Berger & Colella, 1989) on a cartesian grid, or smoothed particles hydrodynamics (SPH; Lucy, 1977; Gingold & Monaghan, 1977; Monaghan, 1992; Springel, 2010b). AMR codes used for star formation simulations such as ORION (e.g. Truelove et al., 1998) have been used successfully to simulate circumstellar disk formation over a few orbital timescales (Kratter et al., 2010). Thanks to mesh refinement, these codes can achieve very high resolution, although the levels of refinement are usually a fixed number, and the dynamical range in density that is achieved is smaller than that of particle-based codes. Similarly, although the higher resolution minimizes the effects of the high-mach-number problem, the grid is still subject to such limitation. On the other hand, SPH – a very popular tool for self-gravitating astrophysical fluid dynamics – offers low-order accuracy for the treatment of contact discontinuities and has poor shock resolution. It also appears to suppress fluid instabilities under certain conditions (Agertz et al., 2007) and suffers from subsonic velocity noise (Abel, 2011). Although SPH is known to require at least a factor of two more resolution elements to achieve similar levels of accuracy in simulations of quiet circumstellar disks (de Val-Borro et al., 2006), it is an attractive tool for simulations of gravitational collapse in disks with radiative cooling (e.g. Meru & Bate, 2010).

Once such alternative is the moving-mesh approach of (Springel, 2010a) (see also

Borgers & Peskin, 1987; Trease, 1988; Dukowicz et al., 1989, for earlier approaches). In this approach, the unstructured grid makes the geometry of the problem irrelevant for the discretization of the equations of hydrodynamics. This moving-mesh method is a finite-volume, Godunov, MUSCL-Hancock scheme with piece-wise linear reconstruction, being thus second-order accurate in space and time. The non-linear step of the MUSCL-Hancock technique (i.e. the solution of the Riemann problem) is carried out in the moving-frame (the instantaneously-at-rest frame) of each cell interface. This provides a method in which a gas disk around a young star can be modeled and evolved independently of the motion of said star, whether it is at rest, moving in a straight line, or orbiting another star: in each of these situations, the numerical scheme would be unaltered and the mesh would evolve according to the motion of the gas.

1.3.2 Beyond Isolated Disks: Disk Evolution and Planet Formation in Young Stellar Clusters

Multiplicity is phenomenon observed early on in stellar evolution (e.g., Mathieu, 1994), suggesting that the primary mechanism for binary formation is tied to the fragmentation of the primordial cloud (e.g., Bate, 2012) or of an early circumstellar disk (e.g., Kratter et al., 2010), and not due to stellar capture at later times when the parent molecular cloud has long dissipated.

Despite the difficulties that binary stars might impose on the dynamics of planets and planet-forming material, planets do form in binary systems: they form around single members of binary stars (e.g. the 55 Cancri system; Butler et al., 1997; Fischer et al., 2008; Dawson & Fabrycky, 2010) as well as in *circum*-binary orbits (e.g. Doyle et

al., 2011; Welsh et al., 2012; Orosz et al., 2012). Of the 720 confirmed planets in the Exoplanet Data Explorer catalog[7] , 79 are associated to a binary system (67 detected by radial velocity measurements and 12 detected through transits). Although most planets and planet candidates known to date orbit around single stars, this could be a selection effect. Statistical analysis suggest that, considering the differences in sample sizes, the fractions of planet-bearing systems among single and multiple stars are indistinguishable (Raghavan et al., 2010), and therefore, binaries could potentially be as likely to host planets as isolated stars (Eggenberger et al., 2004; Bonavita & Desidera, 2007; Mugrauer & Neuhäuser, 2009).

The study of the effects of binary stars on planetary systems has mostly focused on the dynamics of the preexisting planets (Holman & Wiegert, 1999; David et al., 2003; Mudryk & Wu, 2006; Takeda et al., 2008; Moeckel & Veras, 2012; Kratter & Perets, 2012) or of the planetesimals that will form them (e.g. Quintana et al., 2007; Xie et al., 2011) These studies have mostly focused on the dynamics (of planets or planetesimals), but little work has been done on the effects of gas dynamics. Theoretical studies of the gas dynamics of star-disk interaction date back to the seminal paper of Artymowicz & Lubow (1994), which laid the foundations for subsequent studies of tidal truncation of viscous disks and the formation of circumbinary cavities. Disk truncation can be the main reason behind the systematically lower disk-bearing frequency in young low-mass binary components with respect to that of single stars (Bouwman et al., 2006; Monin et al., 2007; Cieza et al., 2009; Kraus et al., 2012). Thus, the presence of a binary outside the disk not only alters the dynamics of disks and planetesimals but it also modifies the gas content in the disk (e.g., Jang-Condell et al., 2008), therefore imposing a strict

[7]http://exoplanets.org

constraint on planet growth and the formation of gas giants.

A Milky Way potentially filled with planets orbiting binaries or individual components of binaries not only opens a series of new questions regarding the dynamical stability of such systems, but it also presents a serious challenge to explaining how these systems came to be in the first place. Furthermore, if planets in these systems form in an analogous way to single-star systems –namely by core accretion in a circumstellar/circumbinary disk– the formation of these primordial gas structures must also be explained. For example, the extreme alignment between the orbit of Kepler-16b and the orbit of the central binary suggests that the entire system formed from the same rotating disk (Winn et al., 2011). However, little is known about the origin of circumbinary disks and their relative frequency in young stellar clusters, as high-resolution direct numerical simulations of star-forming regions are just beginning to form disks self-consistently (Bate, 2012).

Circumbinary disks have been observed in star forming regions. In a detailed census of circumstellar disks in multiple systems in the Taurus-Auriga star forming region, Harris et al. (2012) found a small population of extremely bright sub-mm sources associated to binaries of small separation. This sub-sample includes the disk around GG Tau Aab, one of the most studied circumbinary disks, the most well-known circumbinary disk, GG-Tau (e.g,. Kawabe et al., 1993; Dutrey et al., 1994; Roddier et al., 1996). The existence of circumbinary disks is extremely relevant for the formation of circumbinary planets, but it is also important for the role these disks might play in the evolution of smaller, inner disks around the individual components of the binary by forcing spin-alignment and/or synchronized accretion state (Daemgen et al., 2012).

The characterization of the disks around *individual* stellar components in binaries/multiples is still its early stages, although some conclusions have already been drawn from studies of disk tidal truncation/stripping in binaries and close and intermediate separations. Tidal disk truncation has been studied by Andrews et al. (2010a) and Harris et al. (2012) in the submillimeter in Taurus and by (Daemgen et al., 2012, 2013) in the NIR in Orion. The physical conditions for these two star formation regions are entirely different; Taurus being low mass, and rather quiet, and Orion being massive, dense and disruptive. While disk truncation can be dominated by the tidal effect of companions in Tarus, additional processes like photoevaporation (Mann & Williams, 2009) or disruptive flybys (Olczak et al., 2006) can also be of importance in shaping the sizes of young disks under more violent scenarios like Orion. Although truncation and dispersal processes have been studied in detail in the past, even the simple, long-time tidal truncation of Artymowicz & Lubow (1994) contains uncertainties such as the effective turbulence viscosity of the disk and the effects of mutual inclinations. Furthermore, simple setups like the N-body based truncation model of Pichardo et al. (2005) cannot explain the truncation observed by Harris et al. (2012).

In this thesis, I address the need for more detailed physics and complex three-dimensional geometries in the modeling of circumstellar disks. The additional complexities that these new requirements entail make direct numerical simulation the only alternative to study these systems dynamically well into the non-linear regime. New numerical methods and new powerful supercomputers will play an important role in unveiling the unknown details of how planet formation takes place in multiple stellar systems. Novel methods like the AREPO code suit best for the computational challenges presented by stellar and disk multiplicity.

Part I: Numerical Methods for Moving-Mesh Hydrodynamics

2

The Moving-Mesh Code Arepo

This brief chapter describes the basic equations and properties of the moving-mesh method. Detailed derivations presented in the original AREPO paper are not repeated here. Instead, I chose to emphasize on a few key features of the algorithm and some differences with respect to similar codes.

2.1 The Basics of Moving-Mesh Hydrodynamics

AREPO is built around the idea that one can change the discretization of the domain at every time step. A natural choice of this time-adaptive discretization is to evolve the control volumes or cells in a quasi-Lagrangian fashion, that is, cells move through space as if they were Lagrangian parcels of fluid. To our knowledge, the only way to achieve this, ensuring that cells cover all space and that they deform continuously, is by identifying them as Voronoi tessellation elements.

Besides the novel domain discretization, AREPO includes a second key element which, although related to the moving-mesh concept, is not equivalent nor interdependent with it. This additional element is the implementation of a *boosted* Riemann solver. Without it, AREPO would be much more diffusive. Furthermore, it would be unstable. Before discussing in detail the properties of the boosted Riemann solver, we first derive the discretized finite-volume equation from which AREPO is built.

In conservation-law form, the continuity and momentum equations of gas dynamics are

$$\frac{\partial}{\partial t}\rho + \nabla \cdot [\rho \mathbf{v}] = 0 \quad , \tag{2.1a}$$

$$\frac{\partial}{\partial t}\rho\mathbf{v} + \nabla \cdot \Big[\rho \mathbf{v} \otimes \mathbf{v} + P\mathbf{I} \Big] = 0 \quad , \tag{2.1b}$$

where ρ is the mass density, $\mathbf{v}$ is the velocity field and P is the pressure. The quantity $\mathbf{v} \otimes \mathbf{v}$ denotes the outer product of the velocity with itself (i.e. a rank-2 tensor of components $T_{ab} = v_a v_b$) and $\mathbf{I}$ is the identity matrix.

Finite-volume methods (FVM) make use of these conservation laws in integral form, i.e.

$$\frac{dm_i}{dt} + \int_{V_i} \nabla \cdot [\rho \mathbf{v}] \; dV = 0 \quad ,$$

$$\frac{d\mathbf{p}_i}{dt} + \int_{V_i} \nabla \cdot \Big[\rho \mathbf{v} \otimes \mathbf{v} + P\mathbf{I} \Big] \; dV = 0 \quad ,$$

where m_i and $\mathbf{p}_i$ are the volume integrals of mass density ρ and linear momentum density $\rho\mathbf{v}$ over some control volume V_i (e.g. the i-th cell). Using the Green-Gauss theorem to

convert the volume integral into an area integral we obtain,

$$\frac{dm_i}{dt} = -\sum_{j\neq i} \int_{\partial V = A_{ij}} [\rho \mathbf{v}] \; d\mathbf{A} \quad , \tag{2.3a}$$

$$\frac{d\mathbf{p}_i}{dt} = -\sum_{j\neq i} \int_{\partial V = A_{ij}} \Big[\rho \mathbf{v} \otimes \mathbf{v} + P\mathbf{I} \Big] \; d\mathbf{A} \quad , \tag{2.3b}$$

where an area integral is performed over each of the interfaces that jointly define the boundary of the volume V_i.

In moving-mesh hydrodynamics, control volumes are allowed to vary in time. Therefore, when integrating Equations 4.1 over $V_i = V_i(t)$, the time-dependence of the volume must be taken into account. In such case, we make use of Leibniz-Reynolds transport theorem, which states that, the time-derivative of the volume-integral of any vector quantity $\mathbf{f}$ is given by

$$\frac{d}{dt} \int_{V_i(t)} \mathbf{f} \, dV = \int_{V_i(t)} \frac{\partial \mathbf{f}}{\partial t} \, dV + \int_{\partial V_i(t)} \mathbf{f} \; \mathbf{w} \cdot d\mathbf{A} \quad ,$$

where $\mathbf{w}$ is the velocity vector at the surface ∂V_i of the moving/deforming control volume V_i. This additional term is a surface integral of the same type as those in the right hand side of Equations 2.3, and thus, we can incorporate into the hydrodynamic fluxes of mass and momentum. Therefore, the finite volume equations read instead (Springel, 2010a)

$$\frac{dm_i}{dt} = -\sum_{j\neq i} \int\limits_{\partial V = A_{ij}} [\rho(\mathbf{v} - \mathbf{w})] \; d\mathbf{A} \quad , \tag{3a'}$$

$$\frac{d\mathbf{p}_i}{dt} = -\sum_{j\neq i} \int\limits_{\partial V = A_{ij}} \Big[\rho \mathbf{v} \otimes (\mathbf{v} - \mathbf{w}) + P\mathbf{I} \Big] \; d\mathbf{A} \; . \tag{3b'}$$

Up to this point, Equations 2.3 are still *exact*. In order to solve these ordinary differential equations, FVMs use a time-stepping scheme to integrate the time evolution, as well as a spatial discretization scheme to approximate the area integral at each

interface. In this work, we restrict ourselves to the MUSCL[1] - Hancock technique (van Leer, 1979; see also Quirk, 1994 and Toro, 2009), implemented in AREPO, which uses a linear reconstruction followed by a half time-step time extrapolation to obtain mid-time-step primitive variables at the cell faces. After discretization, the moving-mesh numerical scheme finally takes the form

$$m_i^{(n+1)} = m_i^{(n+1)} - \Delta t \sum_{j \neq i} (\tilde{\mathbf{F}}_m)_{ij}^{(n+1/2)} \cdot \hat{\mathbf{n}}_{ij} A_{ij} \quad , \tag{2.4a}$$

$$\mathbf{p}_i^{(n+1)} = \mathbf{p}_i^{(n+1)} - \Delta t \sum_{j \neq i} (\tilde{\mathbf{F}}_{\mathbf{p}})_{ij}^{(n+1/2)} \hat{\mathbf{n}}_{ij} A_{ij} \quad , \tag{2.4b}$$

where the Godunov fluxes for mass $(\tilde{\mathbf{F}}_m)_{ij}^{(n+1/2)}$ (a vector) and momentum $(\tilde{\mathbf{F}}_{\mathbf{p}})_{ij}^{(n+1/2)}$ (a matrix) are numerical estimates (surface averages over the i-j interface) of the real fluxes in Equation 2.3. The details of the MUSCL-Hancock method used to estimate the time-centered quantities $(\tilde{\mathbf{F}}_m)_{ij}^{(n+1/2)}$ are described in Springel (2010a) and reproduced again in Chapter 3 of this thesis (Muñoz et al., 2013), and we will not repeat them here. Up to this point, the only difference with conventional FVMs is the inclusion of the interface velocity in the definition of the Godunov flux in Equation 2.4.

As explained in Springel (2010a) and analyzed in further detail by Pakmor et al. (2011), the flux of a conserved quantity Q_i through a moving interface can be separated into a static-interface term, and an advection flux correction. From Equation 2.3:

$$\begin{aligned} \frac{dQ_i}{dt} &= -\sum_{j \neq i} \int_{\partial V = A_{ij}} [\rho(\mathbf{v} - \mathbf{w})] \, d\mathbf{A} \\ &= -\sum_{j \neq i} \left[\mathbf{F}_{ij}^{\text{static}} - \mathbf{F}_{ij}^{\text{moving}} \right] \quad . \end{aligned} \tag{2.5}$$

[1]Monotonic Upstream Scheme for Conservation Laws

where we have simplified the notation for the sake of argument.

At this point, the calculation of the rest frame Godunov flux $\mathbf{F}_{ij}^{\rm static}$ depends on the numerical method of choice, since the quasi-Lagrangian feature of the scheme is given by the additional advection flux $\mathbf{F}_{ij}^{\rm moving}$, which is a consequence of the moving interfaces. For example, the tessellation code TESS (Duffell & MacFadyen, 2011) uses an approximate state Riemann solver (HLLC) to obtain $\mathbf{F}_{ij}^{\rm static}$. AREPO, on this other hand, performs a more convoluted operation, with substantially different results. Instead of solving the Riemann problem in the lab frame, AREPO "subtracts" the local bulk flow by boosting onto the frame of the face, using an exact (iterative; Toro, 2009) Riemann solver. The exact Riemann solver returns primitive variables in the boosted frame, which can be easily boosted back to the lab frame, where the Godunov fluxes $\mathbf{F}_{ij}^{\rm static}$ are evaluated using their analytic expressions (the terms in square brackets in Equation 2.3). This operation is more than a mere subtlety, and it lies at the core of what Springel (2010a) referred to as "Galilean invariance" of the code.

Although received with some criticism (see Robertson et al., 2010; McNally et al., 2012a) the debate on the so-called Galilean invariance seems to be of a semantic nature. Indeed, in Robertson et al. (2010) the authors frame the discussion around numerical diffusion, pointing out that –provided that a smooth solution exists– Eulerian codes converge at the expected rate derived from truncation error analysis. The authors conclude that errors can be beaten down with resolution and that, if the code *output* does not show "Galilean invariance", it is simply because the truncation error is too large. This implies that, if AREPO produces results that are seemingly independent of the magnitude of Galilean boosts, it is simply owing to smaller advection errors (i.e., it is the truncation error the one that is not Galilean invariant). Thus, the Galilean

invariance of code output appears as a matter of degree, rather than kind.

In Springel (2010a, §3.4), a derivation of the invariance of the scheme is provided, which shows that a consistent change of reference frames for both the state vectors and the hydrodynamic fluxes produces invariant results in the updated state after a time Δt. This calculation assumes two things: that the Riemann solver produces consistent results modulo a Galilean boost in both frames, and that the time step does not depend on which frame is used. Clearly, said derivation can be strictly valid only if the Riemann problem is solved in the frame of the moving face, and if the Courant time-step condition is defined in terms of the signal velocity respect to the local flow, and with not respect to the lab frame. Therefore, a boosted Riemann solver and a comoving time-step criterion are essential in obtaining Galilean-invariant advection errors.

If the Godunov flux $\mathbf{F}_{ij}^{\mathrm{static}}$ in Equation 2.5 is computed in the lab frame, the velocity field on the left and right states of the discontinuity will be large, and the result will be significantly affected by truncation error. In addition, the much shorter Courant time-step imposed by high velocity flows will increase the number of integrations, building up cumulative error quickly. In this case, an iterative solver can certainly alleviate the consistency problem of solving the Riemann problem in different frames. This is because this type of solver is "exact", meaning that the (relative) error tolerance is arbitrary. However, the strict diffusive toll that is imposed by increasing the number of time-steps is unavoidable.

Duffell & MacFadyen (2011) attribute the larger diffusion of TESS to their use of an approximate Riemann solver. Although the exact Riemann solver could improve performance for the reason described above, the main cause of diffusivity in TESS is

simply owing to their use of approximate solve in the lab frame. To prove that the approximate Riemann solver on its own is of secondary importance, I have implemented an HLLC solver in AREPO. Unfortunately, I was not able to make the scheme stable in lab frame coordinates. As discussed in detail by Pakmor et al. (2011), the fluxes $\mathbf{F}_{ij}^{\rm static}$ and $\mathbf{F}_{ij}^{\rm moving}$ in Equation 2.5 nearly cancel each other out. Therefore, for pathological cases –distorted, sheared or rapidly rotating cells– small errors can change the sign of the net mass flux and violate upwinding, making the scheme unstable. However, a clever algebraic manipulation by Pakmor et al. (2011) allows for the boosting of *any* Riemann solver. Figure 2.1 shows a Kelvin-Helmholtz (KH) instability test problem with a smooth initial condition, as described by McNally et al. (2012a). We run the test using the default iterative Riemann solver in and the boosted HLLC solver. We boost the system along the x-direction by $V_{\rm boost} = 0$ and 10. The iterative and the HLLC solvers show differences on small scales, however, the results of a given solver are very consistent for different boost velocities.

It is interesting, however, that AREPO is unstable when a lab frame Riemann solver is used, since Duffell & MacFadyen (2011) did not report any problems in that regard. One must note that TESS and AREPO use different algorithms to drift the mesh in time, and this could have unexplored consequences for the performance of the scheme.

It is true that the role of the moving mesh as an additional "diffusive agent" has not been explored in detail. Although "grid noise" is seldom discussed in the computational astrophysics literature, it could be of greater relevance for moving-mesh methods than for structured grids. Grid noise originates from the finite precision with which cell and face centers are known. This error should start at the round-off level, but spatial extrapolations and non-symmetric face sweeps can accumulate noise to a levels that

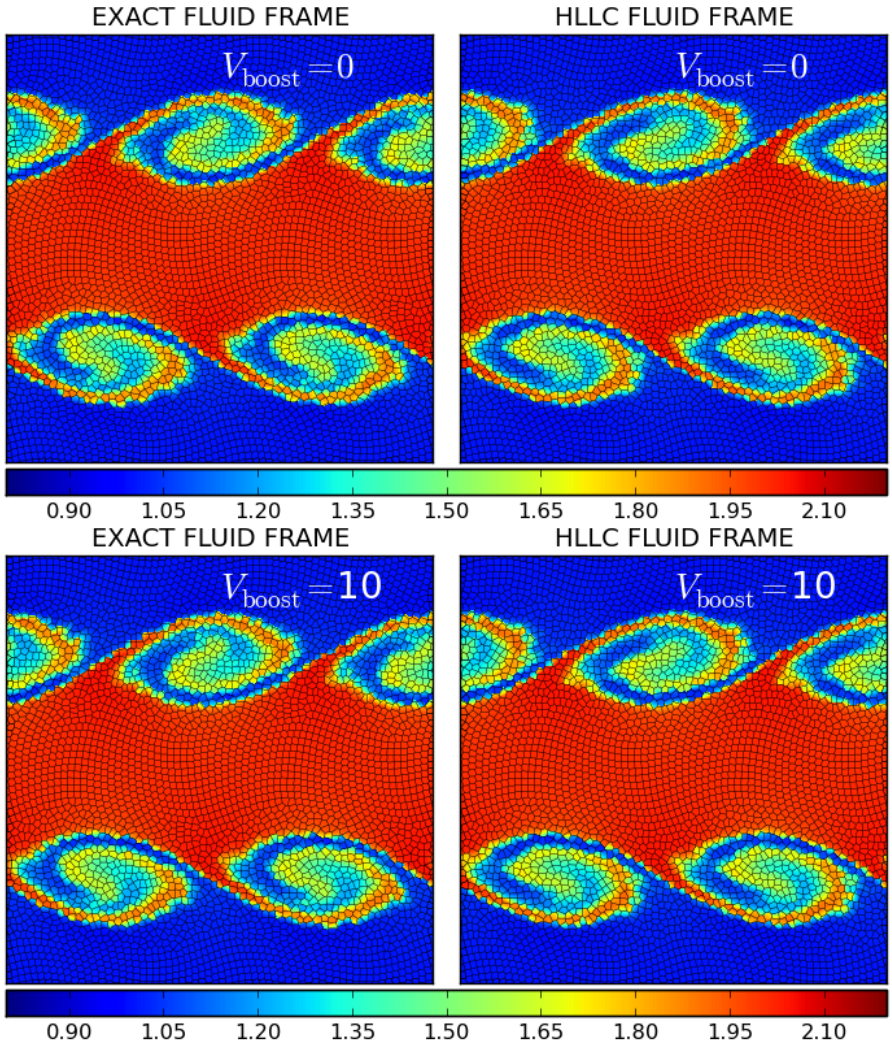


Figure 2.1.— KH instability test for two different Riemann solvers. Top row: KH test as described by McNally et al. (2012a) at a time of $t = 3.0$ using an iterative Riemann solver (left) and an HLLC Riemann solver (right). Bottom row: same as top row but in after boosting the initial condition by $V_{\rm boost} = 10$. Both Riemann solvers are implemented in the frame moving with the cell interfaces.

eventually can affect the code performance. In AREPO, grid noise is not only present, but the mesh-generating points are drifted with a finite precision (normally a leapfrog integrator is used to drift the mesh). Therefore, grid noise can be amplified to levels comparable to the truncation error. Although we have not seen yet a simulation that shows this type of pathological behavior, quantifying the level of grid noise and how it affects the convergence rate of AREPO is worth exploring further.

3

Multi-Dimensional Viscous Flow on a Moving Voronoi Mesh

D. J. Muñoz, V. Springel, M. Vogelsberger & L. Hernquist *The Monthly Notices of the Royal Astronomical Society*, Vol. 428, pp. 254-279, 2013

Abstract

Numerous formulations of finite volume schemes for the Euler and Navier-Stokes equations exist, but in the majority of cases they have been developed for structured and stationary meshes. In many applications, more flexible mesh geometries that can dynamically adjust to the problem at hand and move with the flow in a (quasi) Lagrangian fashion would, however, be highly desirable, as this can allow a significant reduction of advection errors and an accurate realization of curved and moving boundary conditions. Here we describe a novel formulation of viscous continuum hydrodynamics

that solves the equations of motion on a Voronoi mesh created by a set of mesh-generating points. The points can move in an arbitrary manner, but the most natural motion is that given by the fluid velocity itself, such that the mesh dynamically adjusts to the flow. Owing to the mathematical properties of the Voronoi tessellation, pathological mesh-twisting effects are avoided. Our implementation considers the full Navier-Stokes equations and has been realized in the AREPO code both in 2D and 3D. We propose a new approach to compute accurate viscous fluxes for a dynamic Voronoi mesh, and use this to formulate a finite volume solver of the Navier-Stokes equations. Through a number of test problems, including circular Couette flow and flow past a cylindrical obstacle, we show that our new scheme combines good accuracy with geometric flexibility, and hence promises to be competitive with other highly refined Eulerian methods. This will in particular allow astrophysical applications of the AREPO code where physical viscosity is important, such as in the hot plasma in galaxy clusters, or for viscous accretion disk models.

3.1 Introduction

The last two decades have seen remarkable advances in the numerical solution of the compressible Navier-Stokes (NS) equations, which lies at the heart of computational fluid dynamics (CFD) and computational aeroacoustics, but also as numerous applications in astrophysics. In particular, important progress has been made in approaches based on the finite volume method (FVM), both using structured as well as unstructured grids (see Mavriplis, 1997, for a review). Other popular techniques include finite element methods (FEM), discontinuous Galerkin schemes, and even mesh-free approaches such

as smoothed particle hydrodynamics (Sijacki & Springel, 2006).

When unstructured grids have been employed, they were most most often in the form of triangular grids in two dimensions, or tetrahedral grids in three dimensions. Indeed, finite-volume implementations of the two-dimensional NS equations on triangular meshes date back to work by Mavriplis & Jameson (1990), Frink (1994) and Coirier & Powell (1996). Much recent work has also focused on developing optimum mesh-generating algorithms that require minimal human input and yield efficient representations of geometrically complex simulation domains. However, little work has been done on dynamically evolving meshes, such as those we shall consider here.

Because unstructured meshes have been demonstrated to be accurate and efficient for both steady-state and transient compressible inviscid flows (Barth, 1992; Venkatakrishnan, 1996), they are now used regularly in engineering applications. Moreover, the geometric flexibility of unstructured grids allows the use of simple coordinate systems (in the laboratory frame) without the need to work with complex coordinate transformations to describe curved surfaces (e.g. see Toro, 2009). Indeed, hard boundaries can be tailored by carefully positioning a few cell faces or mesh generating points along the surface, and creating the triangulation through Delaunay tessellation. As a result, most NS applications on unstructured meshes for industrial design make use of triangular grids, typically based on the finite element method, although finite volume schemes have also been considered. Detailed reviews and stability analysis of explicit FVM for the NS equations on Cartesian and Delaunay meshes can be found, e.g, in the doctoral theses of Coirier (1994) and Munikrishna (2009).

In this work, we present a numerical scheme that solves the NS equations on a

general unstructured moving mesh that is constructed as the Voronoi tessellation of a distributed set of points that move with the local velocity field. Despite being, in the general sense, an "unstructured" mesh, the Voronoi diagram has a mathematically well-defined structure that makes the resulting schemes comparatively simple and robust (e.g. Mishev, 1998). In fact, this type of mesh is commonly adapted for the construction of finite volume methods for elliptic problems and has been in use in numerical studies of solid state physics (Sukumar et al., 1998; Sukumar, 2009) such as simulations of fractures and cracks (Sukumar & Bolander, 2009), as well as numerical simulations of oil reservoirs. Some studies (Christov, 2009) have also examined how reconstructions designed for unstructured triangulations can be extended to *static* Voronoi meshes.

However, Voronoi meshes have infrequently been applied to hyperbolic conservation laws such as the Euler equations, let alone moving Voronoi meshes. To our knowledge, the earliest attempts to use dynamically adaptive Voronoi tessellations for the NS and Euler equations date back to Borgers & Peskin (1987) and Trease (1988). The former, for very simplified, incompressible, two-dimensional problems, and the latter, a finite-difference method for inviscid three-dimensional flow. Around the same time, Dukowicz et al. (1989) developed the General Topology Godunov Method. This method – based on a mesh that is not quite a Voronoi tessellation, but similar in spirit – was introduced as an alternative to the Lagrangian particle methods (see, for example Brackbill & Monaghan, 1988) which gained increasing popularity in computational plasma physics and astrophysics in the following years.

Recently, a complete three-dimensional implementation of the Euler equations on a moving Voronoi mesh has been described and implemented in the AREPO code by Springel (2010a) (see also Duffell & MacFadyen, 2011). The work we present here is an

extension of the AREPO scheme to the NS equations, which we have realized in this code as an optional module. AREPO can be classified as an arbitrary Lagrangian/Eulerian (ALE; Hirt et al., 1974) code, in the sense that the mesh can be moved with the velocity of the flow so that quasi-Lagrangian behavior results and the mass flux between cells is minimized (although it is not strictly zero, in general). On the other hand, the mesh may also be kept stationary if desired, effectively yielding an Eulerian formulation. We note that because the mesh-generating points may also be arranged on a regular lattice and arbitrarily refined with time, the AREPO code naturally includes ordinary Eulerian techniques on a Cartesian grid and adaptive mesh refinement (AMR) algorithms as special cases.

Besides the work of Duffell & MacFadyen (2011), the new Voronoi-ALE method of Norris et al. (2010), which includes viscous terms, is the approach most closely related to that presented here, although it is restricted to the incompressible NS equations. Also, Ata et al. (2009) have applied a Voronoi-based finite volume scheme to the two-dimensional inviscid shallow water equations, in terms of an algorithm they referred to as the 'natural volume' method.

Although primarily designed for astrophysical fluid dynamics where self-gravity is an important ingredient (see for example Vogelsberger et al., 2012), the moving Voronoi mesh approach of AREPO offers a number of features than can be advantageous for more general problems in fluid dynamics. First, the moving mesh geometry is adaptive in a continuous manner and can naturally respond to the local flow, increasing the resolution automatically and smoothly in regions where the flow converges. (In contrast, AMR codes refine the grid discontinuously in time, which can introduce errors that are potentially difficult to assess.) Importantly, this Lagrangian character of the

dynamics yields reduced advection errors and a very low numerical diffusivity of the scheme. Second, the moving mesh formulation retains the Galilean-invariance of the fluid dynamics at the discretized level of the equations (Springel, 2010a). In other words, the truncation error of the scheme does not depend on the bulk velocity of the system, unlike for traditional Eulerian and AMR codes, and the quality of the solution does not degrade when high-speed flows are present. While conventional fixed-mesh Eulerian codes may, in principle, be able to suppress additional errors from large bulk velocities by using a sufficiently fine mesh (see Robertson et al., 2010, for a study of Galilean invariance in grid codes), this strategy can become computationally prohibitive, and it also depends on the magnitude of the bulk velocity involved. It is therefore desirable to construct efficient methods that yield manifestly Galilean-invariant solutions (modulo floating point round-off errors). Third, the moving mesh approach allows much larger timesteps in the case of rapidly moving flows, because it can avoid the $\Delta t < d/v$ stability constraint (where d is the cell size and v the bulk velocity) that augments the Courant condition in the Eulerian case.

From an astrophysical standpoint, compressible viscous flow remains a viable approximation to more complex or computationally expensive momentum transport mechanisms such as magneto-hydrodynamic turbulence or anisotropic plasma viscosity. Global simulations of cold accretion disks around protostellar objects (e.g. see de Val-Borro et al., 2006) still include shear viscosity coefficients in the form of a Shakura-Sunyaev eddy viscosity coefficient (Shakura & Sunyaev, 1973).

An even clearer case for the need of a viscous treatement of astrophysical gasdynamics is given by the interacluster medium of hot galaxy clusters. Here the Spitzer-Braginskii viscosity (Braginskii, 1965) becomes quite significant, certainly in the

unmagnetized case, which has been studied both using grid (Ruszkowski et al., 2004) and SPH (Sijacki & Springel, 2006) codes. In this regime, the commonly adopted assumption of inviscid behaviour with an effectively infinite Reynolds number is in principle incorrect and should in future simulation work be replaced with a full accounting of the correct physical viscosity.

Additionally, physical viscosity can be implemented on turbulent cascades with resolved inertial range (see Bauer & Springel, 2012, for an application of our viscosity approach) in order to prescribe a well-specified Reynolds number and a physically correct shape for the dissipation range, unaffected by the details of the numerical viscosity of the hydro scheme, which would otherwise induce the dissipation of turbulence on the grid scale. This can in particular inform the ongoing debate whether artificial viscosity effects in SPH can affect the turbulent cascade above the formal resolution length (Bauer & Springel, 2012; Price, 2012).

This paper is organized as follows. In Section 2, we briefly review the basic NS equations we want to solve, and the role and meaning of the different viscosity coefficients. In Section 3, we then introduce in detail our discretization and time integration schemes, emphasizing a description of the calculation of suitable velocity gradient estimates at face centers, and of second-order derivatives of the velocity field. We then move on to discuss the performance of our new approach for a number of test problems in Section 3. Finally, we summarize our results and present our conclusions in Section 4.

3.2 The Navier-Stokes Equations

The compact form of the Euler equations, when written in terms of the vector of conserved quantities $\mathbf{U}$ (Toro, 2009) is

$$\partial_t \mathbf{U} + \nabla \cdot \mathbf{F}_{\text{adv}}(\mathbf{U}) = 0, \tag{3.1}$$

with

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho\mathbf{v} \\ \rho e \end{pmatrix} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e \end{pmatrix}, \tag{3.2}$$

and where

$$\begin{aligned} \mathbf{F}_{\text{adv}}(\mathbf{U}) &= \begin{pmatrix} \rho\mathbf{v} \ , \ \rho\mathbf{v}^T\mathbf{v} + P\mathbf{I} \ , \ (\rho e + P)\mathbf{v} \end{pmatrix} \\ &= \left(\begin{array}{c|c|c} \rho u & 3\times 3 & (\rho e+P)u \\ \rho v & \text{momentum} & (\rho e+P)v \\ \rho w & \text{flux tensor} & (\rho e+P)w \end{array}\right) \end{aligned} \tag{3.3}$$

is the mass-momentum-energy flux density tensor (3×5). The operator $\nabla \cdot (\ )$ in Eq. (4.1) is a tensor divergence, i.e. in tensor notation we have $\{\nabla \cdot \mathbf{F}_{\text{adv}}\}^a = \partial_b F_{\text{adv}}{}^{ba}$. The momentum components in the conservative form of Equation (4.1) represent a transfer of momentum, owing merely to the mechanical transport of different particles of fluid from place to place and to the pressure forces acting on the fluid (e.g. Landau & Lifshitz, 1959). In Eq. (4.1) we have made the advective character of the fluxes explicit by denoting them $\mathbf{F}_{\text{adv}}$.

The internal friction present in any real fluid causes an irreversible transfer of momentum from points where the velocity is large to those where it is small. The momentum flux density tensor is thus altered from its ideal from in Eq. (3.3), where it only contains an inertial and an isotropic component (described by a symmetric stress tensor due to the local pressure P), to a modified expression that accounts for an irreversible viscous transfer of momentum

$$\rho \mathbf{v}^T \mathbf{v} + P\mathbf{I} \longrightarrow \rho \mathbf{v}^T \mathbf{v} + P\mathbf{I} - \mathbf{\Pi}, \tag{3.4}$$

where $P\mathbf{I} - \mathbf{\Pi}$ is the total stress tensor and $\mathbf{\Pi}$ is called the viscous stress tensor. The latter includes the effects of isotropic compression and expansion forces ("bulk viscosity") as well as shearing forces ("shear viscosity").

Similarly, the energy component of Eq. (3.3) is affected by the inclusion of the viscous stress tensor. Because of the dissipative nature of viscosity, a conservative formulation of the NS equations must include a contribution of $\mathbf{\Pi}$ to the energy budget, i.e. the work per unit area per unit time,

$$(\rho e + P)\,\mathbf{v} \longrightarrow (\rho e + P)\,\mathbf{v} - \mathbf{\Pi}\mathbf{v} \tag{3.5}$$

needs to explicitly account for the work done by viscous forces.

A general parametrization of the viscous stress tensor $\mathbf{\Pi}$ is given by

$$\mathbf{\Pi} = \eta \left\{ \left[\nabla \mathbf{v} + (\nabla \mathbf{v})^T \right] - \frac{2}{3}\mathbf{I}\,(\nabla \cdot \mathbf{v}) \right\} + \zeta \mathbf{I}\,(\nabla \cdot \mathbf{v})\,. \tag{3.6}$$

Often, the viscous stress tensor is decomposed into a traceless part and a diagonal part, such that the first corresponds to constant-volume shear deformations (often called the *rate-of-deformation* tensor) and the second to isotropic expansions/contractions.

Accordingly, η in Eq. (3.6) is commonly referred to as the *shear viscosity* and ζ as the *bulk viscosity*. The degree of resistance to uniform contractions/expansions is intrinsic to the molecular/chemical properties of the fluid in question, and can be understood through kinetic theory. In this picture, bulk viscosity arises because kinetic energy of molecules is transferred to internal degrees of freedom. Ideal monoatomic gases (modeled as hard spheres interacting only through elastic collisions) have no internal degrees of freedom, and are thus expected to have vanishing bulk viscosity. At one time Stokes suggested that this might in general be true (the so-called *Stokes' hypothesis* of $\zeta = 0$) but later wrote that he never put much faith in this relationship (Graebel, 2007). Indeed, when deviations from the ideal gas equation of state are included in a hard-sphere, Chapman-Enskog approach to kinetic theory, a non-zero value for the bulk viscosity is obtained. In an extension of the hard sphere fluid model, the Longuet-Higgins-Pople relation $\zeta = (5/3)\eta$ results (March, 2002), motivating the hypothesis that both viscosities are always related in a linear fashion (but see Meier et al., 2005). In general, we consider η and ζ as essentially arbitrary input properties to our simulations, which may also depend on local physical parameters such as temperature or density. Although the effects of physical bulk viscosity are not harder to implement numerically than those of shear viscosity, the physical origin of bulk viscosity is often less clear. Also, we note that many numerical solvers for viscous flow focus on the incompressible regime ($\nabla \cdot \mathbf{v} = 0$), where the existence of a physical bulk viscosity is in any case not of importance. However, for compressible flow, the value of ζ may still become important in certain situations.

When the effects of viscosity are included, the formerly homogeneous differential equations of the Euler form (Eq. 4.1) become

$$\partial_t \mathbf{U} + \nabla \cdot \mathbf{F}_{\text{adv}}(\mathbf{U}) = \mathbf{S}(\mathbf{U}) \tag{3.7}$$

where $\mathbf{S}(\mathbf{U})$ is a viscous source term given by

$$\mathbf{S}(\mathbf{U}) \equiv \nabla \cdot \begin{pmatrix} \mathbf{0} \ , \ \mathbf{\Pi} \ , \ \mathbf{\Pi v} \end{pmatrix} . \tag{3.8}$$

The solution of the Euler equations with source terms is often handled by operator-splitting methods (e.g. Toro, 2009; LeVeque, 2002). That is, the numerical scheme alternates between an advective step that solves the homogeneous part, and a source-term step. Thus, the solution of Eq. (3.7) is split into a two stage problem:

$$\left.\begin{array}{ll} \text{PDE}: & \partial_t \mathbf{U} + \nabla \cdot \mathbf{F}_{\text{adv}}(\mathbf{U}) = 0 \\ \text{IC}: & \mathbf{U}(\mathbf{x}, t) = \mathbf{U}^n \end{array}\right\} \Rightarrow \widetilde{\mathbf{U}}^{n+1} \tag{3.9}$$

$$\left.\begin{array}{ll} \text{ODE}: & \frac{d}{dt}\mathbf{U} = \mathbf{S}(\mathbf{U}) \\ \text{IC}: & \widetilde{\mathbf{U}}^{n+1} \end{array}\right\} \Rightarrow \mathbf{U}^{n+1} \quad . \tag{3.10}$$

Typically, the source terms are more easily written in the primitive variable formulation of the Euler equations. A common choice of the primitive-variable vector is $\mathbf{W} = (\rho, \mathbf{v}, P)^T = (\rho, v_x, v_y, v_z, P)^T$, which we also adopt here. For sources corresponding to the NS viscous terms (Eq. 3.8), only the $\mathbf{v}$ component of $\mathbf{W}$ is affected, thus simplifying the solution method of the source-term step. The three-dimensional Euler equations can be written in the primitive variable form as (Toro, 2009)

$$\partial_t \mathbf{W} + \mathbf{A}_1(\mathbf{W})\, \partial_x \mathbf{W} + \mathbf{A}_2(\mathbf{W})\, \partial_y \mathbf{W} + \mathbf{A}_3(\mathbf{W})\, \partial_z \mathbf{W} = \mathbf{0}. \tag{3.11}$$

For this choice of variables, the coefficient matrices are given by (Toro, 2009)

$$\mathbf{A}_1(\mathbf{W}) = \begin{pmatrix} v_x & \rho & 0 & 0 & 0 \\ 0 & v_x & 0 & 0 & 1/\rho \\ 0 & 0 & v_x & 0 & 0 \\ 0 & 0 & 0 & v_x & 0 \\ 0 & \gamma P & 0 & 0 & v_x \end{pmatrix}, \tag{3.12}$$

$$\mathbf{A}_2(\mathbf{W}) = \begin{pmatrix} v_y & 0 & \rho & 0 & 0 \\ 0 & v_y & 0 & 0 & 0 \\ 0 & 0 & v_y & 0 & 1/\rho \\ 0 & 0 & 0 & v_y & 0 \\ 0 & 0 & \gamma P & 0 & v_y \end{pmatrix}, \tag{3.13}$$

$$\mathbf{A}_3(\mathbf{W}) = \begin{pmatrix} v_z & 0 & 0 & \rho & 0 \\ 0 & v_z & 0 & 0 & 0 \\ 0 & 0 & v_z & 0 & 0 \\ 0 & 0 & 0 & v_z & 1/\rho \\ 0 & 0 & 0 & \gamma P & v_z \end{pmatrix}, \tag{3.14}$$

which is exactly equivalent to the familiar equations

$$\frac{\partial \rho}{\partial t} + \frac{(\partial \rho v_i)}{\partial x_i} = 0 \tag{3.15a}$$

$$\frac{\partial v_i}{\partial t} + v_i \frac{\partial v_i}{\partial x_i} + \frac{1}{\rho}\frac{\partial P}{\partial x_i} = 0 \tag{3.15b}$$

$$\frac{\partial P}{\partial t} + \gamma P \frac{\partial v_i}{\partial x_i} + v_i \frac{\partial P}{\partial x_i} = 0 \quad . \tag{3.15c}$$

In this formulation, the viscous terms of the NS equations, which affect only the velocity,

are (e.g. Landau & Lifshitz, 1959)

$$\mathbf{S}(\mathbf{W}) = \frac{1}{\rho} \begin{pmatrix} 0 \\ \eta \Delta \mathbf{v} + \left(\zeta + \frac{1}{3}\eta\right) \nabla \left(\nabla \cdot \mathbf{v}\right) \\ 0 \end{pmatrix} . \tag{3.16}$$

An alternative to expressing the viscosity effects as source terms is to absorb them directly into the flux divergence,

$$\partial_t \mathbf{U} + \nabla \cdot \left[\mathbf{F}_{\text{adv}} \left(\mathbf{U}\right) - \mathbf{F}_{\text{diff}} \left(\mathbf{U}\right) \right] = 0, \tag{3.17}$$

which highlights the still conservative character of the NS equations. Here diffusive fluxes, defined by

$$\mathbf{F}_{\text{diff}}(\mathbf{U}) = \begin{pmatrix} \mathbf{0} \ , \ \boldsymbol{\Pi} \ , \ \boldsymbol{\Pi}\mathbf{v} \end{pmatrix} , \tag{3.18}$$

are responsible for the effects of viscosity. An implementation of the diffusive fluxes in this conservation-law form is clearly the preferred choice for FVM schemes, which are specifically designed for solving the integral form of these conservation laws. In fact, in this case they *exactly* conserve all the involved quantities to machine precision. We will therefore focus on this method in our study. The central aspect will be the numerical scheme used for estimating the velocity gradients at the cell interfaces, and hence the discretization of the diffusive fluxes. In the next section, we describe our approach for this in detail.

3.3 A Finite Volume Scheme with Viscous Fluxes on a Voronoi Mesh

3.3.1 Basic MUSCL-Hancock Finite Volume Scheme: Overview

Finite volume methods enforce the integral form of the conservation laws on discrete meshes. This approach is manifestly conservative, since fluxes of quantities that leave a cell simply enter the neighboring cell. The NS equations in finite-volume form are

$$\frac{\mathrm{d}\mathbf{Q}_i}{\mathrm{d}t} = -\sum_j A_{ij}\mathbf{F}_{ij} \quad , \quad \text{with} \quad \mathbf{Q}_i = \int_{V_i} \mathbf{U}_i \mathrm{d}V \quad , \tag{3.19}$$

where, in general, the intercell fluxes contain both advective and diffusive contributions,

$$\mathbf{F}_{ij} = \mathbf{F}_{\mathrm{adv},ij} - \mathbf{F}_{\mathrm{diff},ij}. \tag{3.20}$$

The scheme used by AREPO is the finite volume MUSCL-Hancock approach, consisting of a MUSCL (Monotone Upstream-centered Schemes for Conservation Laws) linear reconstruction stage, and a Hancock two-stage time integration

$$\mathbf{Q}_i^{n+1} = \mathbf{Q}_i^n - \Delta t \sum_j A_{ij}\hat{\mathbf{F}}_{ij}^{n+1/2}, \tag{3.21}$$

where the numerical fluxes $\hat{\mathbf{F}}_{ij}^{n+1/2}$ represent appropriately time-averaged approximations to the true flux $\mathbf{F}_{ij}$ across the interface shared by cells i and j. The time label $n + 1/2$ in Eq. (3.21) indicates that an intermediate-stage (a half time-step evolution) has been performed to obtain the numerical estimate of $\mathbf{F}_{ij}$, meaning that the time-stepping in Eq. (3.21) uses time-centered fluxes, giving it second-order accuracy. The Hancock part of the scheme is a two-step approach (the familiar predictor-corrector algorithm) in which the correction half-step is obtained from the solution of the 1-D Riemann problem

across each face of the control volume. The general finite volume MUSCL-Hancock scheme has hence the following three steps (Toro, 2009):

(I) Gradient Estimation, Linear Data Reconstruction and Boundary Value Extrapolation Once a local gradient estimate for the conserved quantities $\mathbf{U}_i = (\rho, \rho\mathbf{v}, \rho e)_i$ of cell i is available, linear data reconstruction takes the form

$$\begin{aligned} \mathbf{U}_{ij}^L &= \mathbf{U}_i + \nabla\mathbf{U}_i^n(\mathbf{f}_{ij} - \mathbf{s}_i) \\ \mathbf{U}_{ij}^R &= \mathbf{U}_j + \nabla\mathbf{U}_j^n(\mathbf{f}_{ji} - \mathbf{s}_j) \end{aligned} \tag{3.22}$$

where we denote by $\mathbf{U}_{ij}^L$ the estimated vector of conserved variables at the centroid of the ij-interface, obtained by linearly extrapolating the cell-centered values $\mathbf{U}_i$ of the i-th cell (on the "left" side) from $\mathbf{s}_i$, the cell's center position, to $\mathbf{f}_{ij}$. Similarly, $\mathbf{U}_{ij}^R$ corresponds to the estimates of the face-centroid values obtained by linear extrapolation of the cell-centered values of the j-th cell (the "right" side), whose center position is $\mathbf{s}_j$. In both cases, $\mathbf{f}_{ij} = \mathbf{f}_{ji}$ is the position vector of the face centroid between the cells. The Jacobian $\nabla\mathbf{U}_i^n$ is explicitly labeled with superscript n to point out that it corresponds to the estimate of spatial derivatives at the beginning of the time-step.

(II) Evolution of Boundary Extrapolated Values This is, strictly speaking, the "predictor" half time-step. The conserved variables are evolved for $\Delta t/2$ with flux estimates obtained from the values at the beginning of the time-step:

$$\begin{aligned} \widehat{\mathbf{U}}_{ij}^L &= \mathbf{U}_{ij}^L - \frac{\Delta t}{2}\frac{1}{V_i}\sum_j A_{ij}\mathbf{F}_{ij}^n \\ \widehat{\mathbf{U}}_{ij}^R &= \mathbf{U}_{ij}^R - \frac{\Delta t}{2}\frac{1}{V_j}\sum_j A_{ij}\mathbf{F}_{ji}^n \end{aligned} \tag{3.23}$$

(III) Solution of 1-D Riemann Problems and Computation of Godunov Fluxes This corresponds to the "corrector" half time-step in the two-stage Hancock approach. Once the values to the right and left of the interface at time $\Delta t/2$ are known, the discontinuity is treated as a one-dimensional Riemann problem. An exact or approximate Riemann solver is used to return values of ρ, $\rho\mathbf{v}$ and ρe at the interface, at a time corresponding to $n + 1/2$. From these values, the advective fluxes can be directly computed (Eq. 3.3). These are time-centered fluxes $\mathbf{F}_{ij}^{n+1/2}$ used to update the system from the beginning of the time-step to its end,

$$\mathbf{U}_{ij}^{n+1} = \mathbf{U}_{ij}^{n} - \Delta t \frac{1}{V_i} \sum_j A_{ij} \mathbf{F}_{ij}^{n+1/2}. \tag{3.24}$$

Figures 3.1 and 3.2 illustrate the mesh geometry and the basic steps of this inviscid numerical scheme implemented in AREPO. One additional point we have not explicitly discussed here for simplicity is the treatment of the mesh motion, as indicated in Fig. 3.2. This is incorporated into the scheme by evaluating all fluxes in the rest frame of the corresponding face, as described by Springel (2010a). This requires appropriate boosts of the fluid states and the fluxes from the lab frame to the rest frame of each face, and back. For a Voronoi mesh, the face velocities are fully specified by the velocities of all the mesh generating points. The latter can be chosen freely in principle, but if they are set equal to the fluid velocities of the corresponding cells, a Lagrangian behavior and a manifestly Galilean-invariant discretization scheme is obtained in which the truncation error does not depend on the bulk velocity of the system.

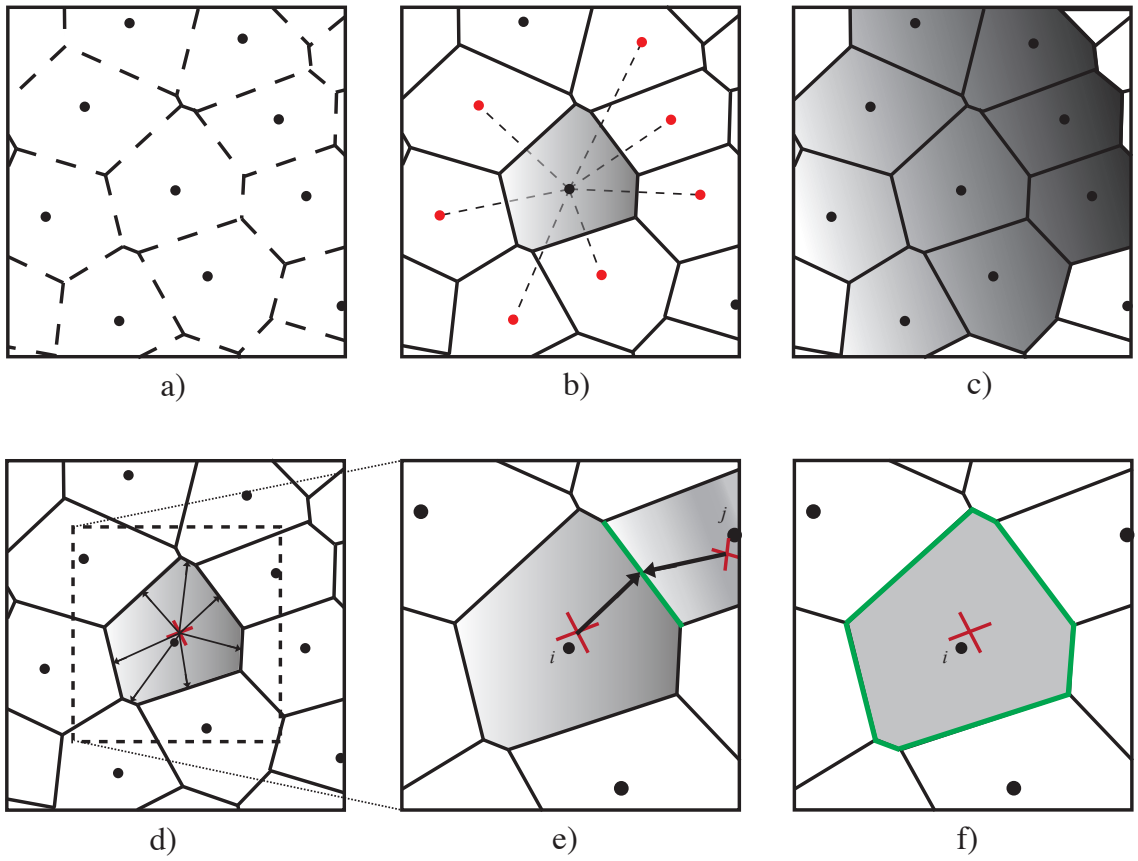


Figure 3.1.— Schematic representation of the mesh geometry and the MUSCL-Hancock integration scheme implemented in AREPO: a) TheVoronoi mesh is uniquely determined by the location of the mesh-generating points. b) A gradient estimate for all primitive variables is obtained from the immediate neighbors of a given cell. c) The gradient-estimation process is repeated for each cell in the domain and thus a piece-wise linear reconstruction is obtained for each primitive variable. d) The primitive variables are extrapolated toward each interface and evolved for half a time-step. e) For each face, a pair of extrapolated quantities for two neighboring cells i and j forms a local Riemann problem. f) The Riemann problem is solved for each face of a cell, yielding time-centered Godunov fluxes for the entire boundary of the control volume V_i of cell i. These fluxes are used for updating the conserved quantities of the cell through Eq. (3.21).

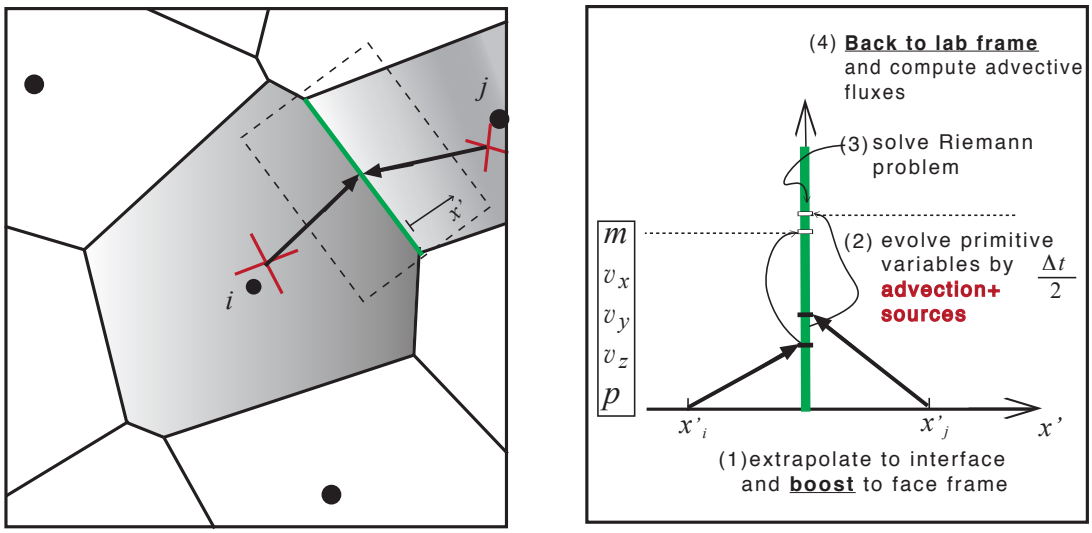


Figure 3.2.— Detailed description of the flux calculation with a Riemann solver in step e) of Fig. 3.1. For the case of a moving mesh, the standard MUSCL-Hancock method needs to be augmented with Galilean-boosts, as described by Springel (2010a): (1) The extrapolation towards each interface is followed by a Galilean boost of the velocities to the rest frame of the face, and by a rotation of the coordinate axes. Each face is then treated as a one-dimensional discontinuity. Thus, the axes are oriented in the rotated frame such that the x'-axis coincides with the normal to the face (left panel). (2) The primitive variables in the moving frame are evolved for half a time-step, including source terms if present (e.g. gravity or viscosity). (3) A one dimensional Riemann problem is solved at the interface. (4) The velocities are translated back to the lab frame and the advective fluxes are computed.

3.3.2 A MUSCL-Hancock Finite-Volume Scheme with Viscous Terms

A cell-centered, finite-volume solution of the NS equation can be written as

$$\mathbf{Q}_i^{n+1} = \mathbf{Q}_i^n - \Delta t \sum_j A_{ij} \hat{\mathbf{F}}_{\mathrm{adv},ij}^{n+1/2} - \Delta t \sum_j A_{ij} \hat{\mathbf{F}}_{\mathrm{diff},ij}^{n+1/2}, \tag{3.25}$$

where we have retained the distinction between advective and viscous fluxes. As in the case of the Euler equations, the numerical method essentially consists of the problem of finding accurate time-centered numerical fluxes across each of the interfaces of a cell. How to do this in detail for the diffusive part of the fluxes has been the focus of numerous efficiency and stability analyses (see Puigt et al., 2010, for a detailed description).

Eq. (3.25) uses time-centered fluxes, obtained here with the two-step Hancock technique, as described above. Thus, for estimating both $\hat{\mathbf{F}}_{\mathrm{adv},ij}^{n+1/2}$ and $\hat{\mathbf{F}}_{\mathrm{diff},ij}^{n+1/2}$ a half time-step predictor stage is required. In the MUSCL-Hancock approach for inviscid flow, this step is carried out by linear reconstruction from each cell center to the interface, followed by solving a one-dimensional Riemann problem at the interface where the extrapolations meet. The traditional formulation of the Riemann problem and its solution are exclusive to hyperbolic differential equations and thus do not provide exact solutions for the NS equations. Since a general solution for the viscous Riemann problem does not exist, we will treat the viscous fluxes in Eq. (3.25) as a correction to the solution of an otherwise inviscid flow.

Our NS version of the MUSCL-Hancock scheme consists of the following three different stages (in addition to those described in Section. 3.3.1):

(A) Correct the MUSCL linear extrapolation of primitive variables by applying a

viscous kick.

(B) Extrapolate the cell-centered gradients linearly and evolve them for half a time-step.

(C) Average the extrapolated velocity gradients at the interface and use them to estimate viscous fluxes.

To extrapolate the gradients from their cell-centered values to the interfaces, information about the higher-order derivatives of the primitive variables is needed. If gradients are assumed to vary linearly in space, an estimator for the Hessian matrix for each of the five primitive variables is sufficient. Evidently, enough information is contained in the cell-centered quantities to estimate both the local gradient $\nabla\phi$ and the Hessian $\mathbf{H}^{\phi}$ corresponding to a given scalar quantity ϕ. However, estimating both of these simultaneously is significantly more difficult than estimating them one after the other. Therefore, we will effectively treat ϕ and $\nabla\phi$ as two independent fields that vary linearly in space, and this variation needs to be estimated from the mesh data through a suitably discretized differential operator.

As a simpler alternative to the gradient reconstruction approach, we briefly describe how one can use the gradients already available from the linear reconstruction step. In this approximation, a given quantity varies only linearly within the control volume, such that consistently evaluated gradients are piece-wise constant. This means that each interface represents a discontinuity in the gradient field $\nabla\phi$. Naively, one may think that the arithmetic average of both gradients that meet at a face is a good estimate for the gradient at the interface itself. However, on second thought, one realizes that both cells do no necessarily have the same weight if cells of different volume meet. Furthermore, the unweighted average of the two cell-centered values really represents the value at

the midpoint of the two mesh-generating points, which, for a Voronoi mesh, can be substantially offset from the mid-point of the face. We therefore adopt the approach of Loh (2007), which consists in choosing one of the two gradients that meet at the interface, based on prior knowledge of the direction of the flow across the interface. Thus the three-stage scheme introduced above could be alternatively replaced by the simpler method:

(A'- C') At the cell interface where two different gradients meet, choose the upwind gradient.

In either method, once we have an estimate of both viscous and advective fluxes, the time-step evolution of the conserved quantities $\mathbf{Q}_i$ is carried out as in Eq. (3.25). However, the approach (A-C) is preferable to the Loh (2007) scheme because it uses time-centered estimates for both $\hat{\mathbf{F}}^{n+1/2}_{\mathrm{adv},ij}$ and $\hat{\mathbf{F}}^{n+1/2}_{\mathrm{diff},ij}$, hence preserving the order of accuracy of the original inviscid scheme. We therefore now provide a more detailed description of the individual steps in this three-stage approach.

(A) Viscosity Kicks

Although Eq. (3.25) is written in an unsplit form, the predictor step is indeed operator split, evolving the advective and diffusive terms separately (e.g. Coirier & Powell, 1996). While our method for estimating the advective fluxes remains the MUSCL-Hancock scheme, the technique for estimating the diffusive fluxes is essentially contained in the estimation of the velocity gradients at each interface (see Coirier, 1994; Puigt et al., 2010, for a series of tests on different interface gradient estimates). Looking for better accuracy, we have chosen to couple these two otherwise independent procedures by

correcting/biasing the linear extrapolation of the velocity field (stage (I) in Section 3.3.1) with a viscous source term.

The benefit of carrying out a linear extrapolation to cell interfaces in primitive variables is the simplicity of the Galilean transformation needed to boost the quantities to the frame of a moving interface. Since the Galilean boost does not affect the mass and pressure of a given cell, only the local velocity field is transformed. In addition, adding force source terms to the equations of motion in primitive variable formulation is simpler, since these only couple to the momentum equations. Thus, a "viscous kick" can be applied to the velocity field in the half time-step evolution stage:

$$\Delta \mathbf{v}_{\text{visc}} = \frac{\Delta t}{2}\left[\frac{\eta}{\rho}\nabla^2\mathbf{v} + \frac{\zeta + \frac{1}{3}\eta}{\rho}\nabla\left(\nabla\cdot\mathbf{v}\right)\right]. \tag{3.26}$$

In this way, the subsequent linear extrapolation of primitive variables will already include viscosity effects to first order in time.

While working with numerical fluxes across interfaces requires velocity gradients, the use of cell-centered source terms in Eq. (3.26) calls for second order derivatives of the velocity field. Thus, in addition to the cell-centered velocity gradients ∇v_x, ∇v_y and ∇v_z, the cell-centered Hessian matrices $\mathbf{H}^{v_x}$, $\mathbf{H}^{v_x}$ and $\mathbf{H}^{v_x}$ are now needed. As we will see below, these matrices will be of use in more than one occasion, justifying the computational cost incurred to calculate them.

(B) Linear Extrapolation of Gradients

The linear reconstruction implemented in our MUSCL-Hancock approach essentially assumes that the gradient of a scalar quantity ϕ does not vary significantly across the spatial scale of a cell. For smooth flows, the gradients of two neighboring cells $\nabla\phi\Big|_i$ and

$\nabla\phi\Big|_j$ will not differ significantly. Furthermore, in the presence of strong discontinuities, gradients on each side will be slope-limited, and therefore will not differ by much either. Hence, a first guess for the gradient at the interface between two cells is just the average of the cell-centered estimates at each side of the face

$$\widetilde{\nabla\phi}\Big|_{ij} = \frac{\langle\nabla\phi\rangle_i + \langle\nabla\phi\rangle_j}{2}. \tag{3.27}$$

However, as we pointed out earlier, the gradient average above is actually representative of the midpoint between the two cell centers $\mathbf{r}_i$ and $\mathbf{r}_j$, which in general does not lie close to the center of the face in a Voronoi mesh, and may in fact lie within a third cell. Unless gradients are assumed to vary within a cell, it will not be possible to assign the estimate to the center of the interface with any confidence.

Let us assume that the scalar field $\phi(\mathbf{r})$ is infinitely differentiable and, consequently, so is its first derivative. Thus, we can Taylor expand both quantities to arbitrary order around a mesh generating point $\mathbf{r}_0$:

$$\begin{aligned}\phi(\mathbf{r}) =& \phi(\mathbf{r}_0) + \nabla\phi\Big|_{\mathbf{r}_0}(\mathbf{r}-\mathbf{r}_0) \\ &+ \frac{1}{2}(\mathbf{r}-\mathbf{r}_0)^T\mathbf{H}^\phi\Big|_{\mathbf{r}_0}(\mathbf{r}-\mathbf{r}_0) + \mathcal{O}(\mathbf{d}^3)\end{aligned} \tag{3.28}$$

$$\begin{aligned}\nabla\phi(\mathbf{r}) =& \nabla\phi\Big|_{\mathbf{r}_0} + \mathbf{H}^\phi\Big|_{\mathbf{r}_0}(\mathbf{r}-\mathbf{r}_0) \\ &+ \frac{1}{2}(\mathbf{r}-\mathbf{r}_0)^T\mathbf{D}^\phi\Big|_{\mathbf{r}_0}(\mathbf{r}-\mathbf{r}_0) + \mathcal{O}(\mathbf{d}^3)\end{aligned} \tag{3.29}$$

where $\mathbf{H}^\phi$ is the Hessian matrix of the scalar quantity ϕ and $\mathbf{D}^\phi$ is a $3\times3\times3$ tensor containing the third-order derivatives of ϕ (i.e. $D_{abc} = \partial^3\phi/\partial x_a\partial x_b\partial x_c$). Truncating both Taylor expansions to first-order in $\mathbf{d} = \mathbf{r}-\mathbf{r}_0$, we see that we can obtain linear reconstructions for both the physical quantities and their gradients provided that we have numerical estimates for both the gradients and the Hessians at each mesh generating point.

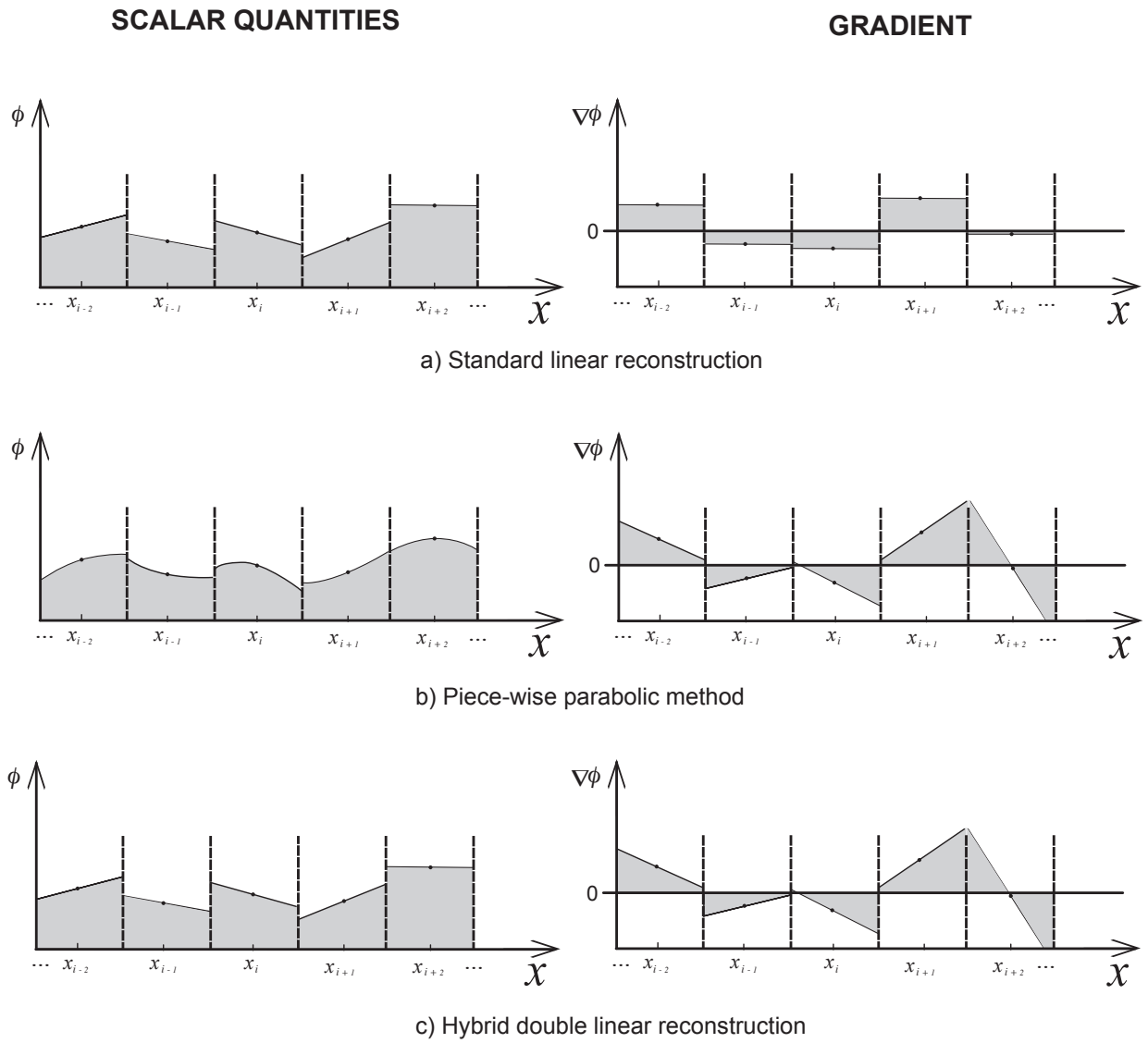


Figure 3.3.— Schematic representation of the double linear reconstruction proposed in this work compared to standard linear reconstruction and parabolic reconstruction.

We emphasize that a Taylor expansion is not equivalent to a polynomial data reconstruction. Indeed, it is desirable that reconstruction schemes are manifestly conservative, in the sense that the average of the reconstruction over the cell should be identical to the value of ϕ at the geometric center of the cell. This property of reconstruction schemes is sometimes referred to as K-exactness, meaning that if a polynomial reconstruction is cell-averaged over the mesh, the reconstruction procedure recovers the same polynomial. This condition is trivially satisfied for a linear reconstruction of the form $\phi(\mathbf{r}) = \phi_i + \langle\nabla\phi\rangle_i(\mathbf{r}-\mathbf{s}_0)$. However, higher-order reconstruction schemes require the use of zero-mean polynomials, which, beyond first-order, differ from the Taylor series (e.g. Colella & Woodward, 1984; Coirier & Powell, 1996).

The linear reconstruction of the scalar field ϕ and of the vector field $\nabla\phi$, treated as if they were independent quantities, effectively constitutes a hybrid method between standard linear reconstruction and fully K-exact second-order reconstruction, as illustrated in Figure 3.3. In this approximation, second derivatives are considered negligible for the spatial reconstruction of the primitive quantities, but they are still included for a more accurate estimate of the gradients near the cell interfaces. We also note, that in this way our numerical scheme reduces to that originally in AREPO (which is second-order-accurate) when the viscous fluxes are disabled.

Once an estimate for the Hessian matrix $\mathbf{H}^{\phi}\Big|_{\mathbf{r}_0}$ is available (Section 3.3.3), a linear extrapolation of the gradients from the cell centers to the interfaces can be obtained from

$$\widetilde{\nabla\phi}\Big|_{ij} = \langle\nabla\phi\rangle_i + \langle\mathbf{H}^{\phi}\rangle(\mathbf{f}_{ij} - \mathbf{r}_i), \tag{3.30}$$

which is a better approximation than Eq. (3.27). However, the time evolution of the gradients during a single step could be equally important as their spatial variation over

the length scale of a cell, hence we also need to evolve them for half a time-step to obtain a time integration scheme that is consistent with the second-order accurate two-stage MUSCL-Hancock approach. In the latter, to extrapolate and evolve a scalar quantity ϕ we consider

$$\phi\Big|_{ij} = \phi_i + \nabla\phi\Big|_{\mathbf{r}_0} (\mathbf{f}_{ij} - \mathbf{s}_i) - \frac{\Delta t}{2} \left\langle \frac{\partial \phi}{\partial t} \right\rangle_i \tag{3.31}$$

where the time derivative of the quantity ϕ in the control volume of the i-th cell can be obtained from the primitive variable formulation of the Euler equations in tensor notation:

$$\partial_t W_\alpha + A_{\alpha\beta b}(\mathbf{W}) \partial_b W_\beta = 0. \tag{3.32}$$

Here sums over repeated indices are implied. Latin indices $a, b, c, d...$ take the values $1, 2, 3$ or x, y, z, while Greek indices $\alpha, \beta, \gamma, ...$ take the values $1, 2, 3, 4, 5$ and are used to number the components of the primitive quantity vector ($W_\alpha = \rho, v_x, v_y, v_z, P$ for $\alpha = 1, 2, 3, 4, 5$, respectively). As with our previous notation, the indices i, j and k are reserved for labeling the mesh generating points and their associated cells.

Eq. (3.32) is an advection equation for the primitive variables. Analogously, to "advect" the gradients of the primitive variables from the cell center to the interface, we can ignore the viscous terms and derive an equation of motion for the spatial derivatives by differentiating Eq. (3.32):

$$\partial_a \partial_t W_\alpha + (\partial_a A_{\alpha\beta b})\, \partial_b W_\beta + A_{\alpha\beta b} \partial_a \partial_b W_\beta = 0, \tag{3.33}$$

where we can identify the Jacobian matrix of the primitive variables as $J_{\alpha a} \equiv \partial_a W_\alpha = W_{\alpha,a}$, and the Hessian tensor ($5 \times 3 \times 3$) of the primitive variables as $H_{\beta ba} \equiv \partial_b \partial_a W_\beta = W_{\beta,b,a}$. Therefore, the time derivative of each component of the

primitive variable Jacobian matrix is

$$\partial_t J_{\alpha a} = B_{\alpha\beta b a} J_{\beta b} - A_{\alpha\beta b} H_{\beta b a}, \tag{3.34}$$

where we introduced the rank-4 tensor $B_{\alpha\beta ba} \equiv \partial_a A_{\alpha\beta b} = A_{\alpha\beta b,a}$. Since $A_{\alpha\beta b}$ is a function of the primitive variables W_α, the tensor $B_{\alpha\beta ba}$ can also be written as (see Appendix)

$$B_{\alpha\beta ba} = \frac{\partial A_{\alpha\beta b}}{\partial W_\gamma} \partial_a W_\gamma, \tag{3.35}$$

and therefore its numerical estimate is given by the product of the exact derivatives $\partial A_{\alpha\beta b}/\partial W_\gamma$ (evaluated with values of the primitive variables at the center of the cell) and the (already available) numerical estimates for the gradients $\partial_a W_\gamma = J_{\gamma a}$. The second term on the right hand side of Eq. (3.34) is the product of the known coefficients $A_{\alpha\beta b}$ (evaluated at the center of the cell) and the numerical estimates of the Hessian tensor $H_{\beta ba}$.

Finally, with a numerical estimate of $H_{\beta ba}$ at hand (see Section 3.3.3), the extrapolated and half time-step evolved gradients of the velocity are (in analogy to Eq. 3.31):

$$\nabla v_x \Big|_{ij} = \langle \nabla v_x \rangle_i + \langle \mathbf{H}^{v_x} \rangle_i \left(\mathbf{f}_{ij} - \mathbf{s}_i \right) + \frac{\Delta t}{2} \left\langle \frac{\partial \nabla v_x}{\partial t} \right\rangle_i, \tag{3.36}$$

with analogous expressions for $\nabla v_y \,|_{ij}$ and $\nabla v_z \,|_{ij}$. In Eq. (3.36), the term $\langle \partial \nabla v_x / \partial t \rangle_i$ is obtained from Eq. (3.34) with $\alpha = 2$ and $a = 1, 2, 3$.

In Fig. 3.4, we show a sketch of the different steps involved in obtaining time-centered diffusive fluxes. We point out that taking the Hessian matrices of the velocity field to be identically zero is *not* equivalent to the alternative scheme (A'). The third term to the right hand side of Eq. (3.36) is still different from zero even if $H_{\beta ba} = 0$ (Eq. 3.34) since, in general, $B_{\alpha\beta ba} J_{\beta b} \neq 0$. By advecting the gradients according to Eq. (3.34) we gain

additional accuracy at no additional computational expense because the terms $B_{\alpha\beta ba}J_{\beta b}$ are known exactly (see Appendix), given the values of the primitive variables and their respective gradients at the center of each cell.

(C) Viscous Flux Calculation

An accurate estimate of the viscous fluxes between two cells requires an accurate estimate of the velocity gradients at the interface. The gradient extrapolation method described above produces in general two different values of the velocity gradient that meet at the interface. This defines a general Riemann problem for the differential equation in Eq. (3.34) which is no longer a homogeneous hyperbolic differential equation. Therefore, attempting to solve this new Riemann problem for the spatial derivatives of the scalar quantities introduces a significant additional difficulty. For simplicity, we will assume that the differences between two gradient extrapolations meeting at an interface are small enough such that a simple arithmetic mean can be used. This assumption, of course, is valid only when the field of second derivatives is sufficiently smooth (see Section 3.3.3).

The time and area averaged flux across the face i-j that moves with speed $\mathbf{w}$ is defined as

$$\begin{aligned}\hat{\mathbf{F}}_{ij} = \frac{1}{\Delta t}\frac{1}{A_{ij}}\int_{\Delta t}\int_{A_{ij}} \Big[\, \mathbf{F}_{\text{adv}}(\mathbf{U}) - \mathbf{U}\mathbf{w}^T \\ -\mathbf{F}_{\text{diff}}(\mathbf{W}, \partial\mathbf{W}/\partial\mathbf{r}) \Big]\, \mathrm{d}\mathbf{A}_{ij}\, \mathrm{d}t \\ \equiv \hat{\mathbf{F}}_{\text{adv},ij} - \hat{\mathbf{F}}_{\text{diff},ij} \quad .\end{aligned} \tag{3.37}$$

The numerical or Godunov estimate of these fluxes is chosen so that the analytic expressions for $\mathbf{F}_{\text{adv}}(\mathbf{U})$ and $\mathbf{F}_{\text{diff}}(\mathbf{W}, \partial\mathbf{W}/\partial\mathbf{r})$ are evaluated with numerical estimates of

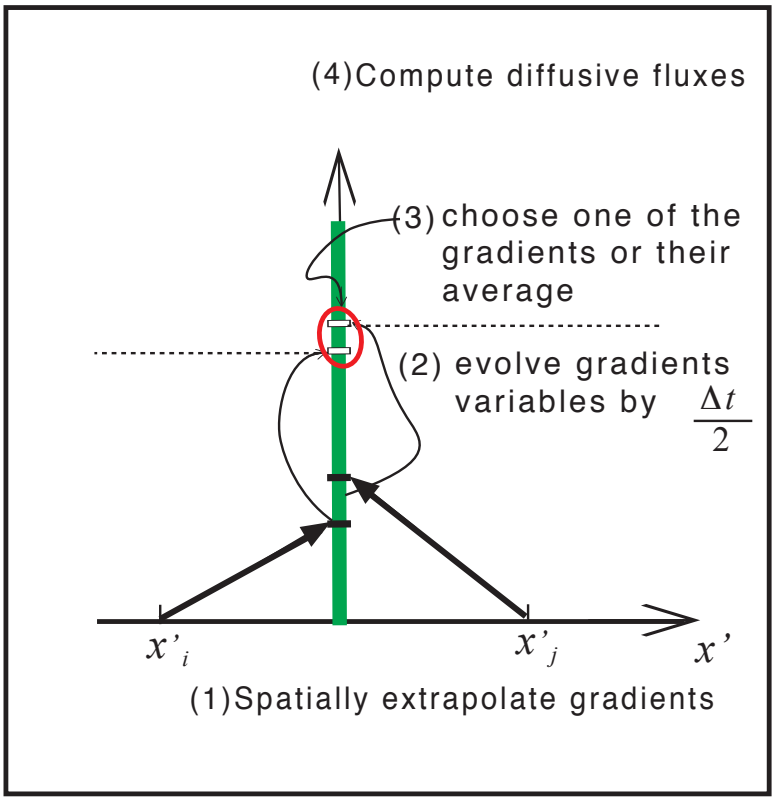


Figure 3.4.— Sketch illustrating the individual steps involved in the extrapolation and half time-step evolution of the gradients, analogous to the advective flux calculation shown in Fig. 3.2. The different steps are: (1) spatial extrapolation of the gradients, followed by (2) a time advance by $\Delta t/2$ according to Eq. (3.31), and (3) an approximate evaluation right at the interface. In step (4), the viscous fluxes are determined by evaluating Eq. (3.18) with the values of the primitive variables and the velocity gradients at the interface.

$\mathbf{U}$, $\mathbf{W}$ and $\partial\mathbf{W}/\partial\mathbf{r}$ at the centroid of the interface. The advective Godunov fluxes are

$$\hat{\mathbf{F}}_{\mathrm{adv},ij} = \left[\mathbf{F}_{\mathrm{adv}}(\mathbf{U}^{\mathrm{lab}}_{\mathrm{Riem}}) - \mathbf{U}^{\mathrm{lab}}_{\mathrm{Riem}}\mathbf{w}^T\right]\hat{\mathbf{n}}_{ij}, \tag{3.38}$$

where $\mathbf{U}^{\mathrm{lab}}_{\mathrm{Riem}}$ is the conserved variable vector at the centroid of the interface, as seen in the lab frame, obtained from the solution of a 1-D Riemann problem across the i-j interface and along its normal. Multiplying by $\hat{\mathbf{n}}_{ij}$ is equivalent to projecting the flux matrix $\mathbf{F}_{\mathrm{adv}}$ (Eq. 3.3) along the normal of each face. The Godunov fluxes $\hat{\mathbf{F}}_{\mathrm{adv},ij}$ and $\hat{\mathbf{F}}_{\mathrm{diff},ij}$ are thus 5-component vectors. The diffusive Godunov flux vector is obtained from the diffusive flux 5×3 matrix

$$\mathbf{F}_{\mathrm{diff}} = \begin{bmatrix} 0 & 0 & 0 \\ \Pi_{xx} & \Pi_{yx} & \Pi_{zx} \\ \Pi_{xy} & \Pi_{yy} & \Pi_{zy} \\ \Pi_{xz} & \Pi_{yz} & \Pi_{zz} \\ v_x\Pi_{xx} + v_y\Pi_{xy} + v_z\Pi_{xz} & v_x\Pi_{yx} + v_y\Pi_{yy} + v_z\Pi_{yz} & v_x\Pi_{zx} + v_y\Pi_{zy} + v_z\Pi_{zz} \end{bmatrix} \tag{3.39}$$

where Π_{ab} are the components of the viscous stress tensor $\mathbf{\Pi}$, which depend on the local value of the velocity and the velocity gradients. These components are:

$$\begin{aligned}
\Pi_{xx} &= \frac{4}{3}\eta\partial_x v_x - \frac{2}{3}\eta\left(\partial_y v_y + \partial_z v_z\right) + \zeta\nabla\cdot\mathbf{v} \\
\Pi_{yy} &= \frac{4}{3}\eta\partial_y v_y - \frac{2}{3}\eta(\partial_z v_z + \partial_x v_x) + \zeta\nabla\cdot\mathbf{v} \\
\Pi_{zz} &= \frac{4}{3}\eta\partial_z v_z - \frac{2}{3}\eta(\partial_x v_x + \partial_y v_y) + \zeta\nabla\cdot\mathbf{v} \\
\Pi_{xy} &= \Pi_{yx} = \eta\left(\partial_y v_x + \partial_x v_y\right) \\
\Pi_{yz} &= \Pi_{zy} = \eta\left(\partial_z v_y + \partial_y v_z\right) \\
\Pi_{zx} &= \Pi_{xz} = \eta\left(\partial_x v_z + \partial_z v_x\right)
\end{aligned} \tag{3.40}$$

Just like with the advective fluxes, the flux tensor (Eq. 3.39) must be projected onto the normal $\hat{\mathbf{n}}_{ij}$ of each ij-interface to obtain the 5-component vector

$$\hat{\mathbf{F}}_{\mathrm{diff},ij} = \mathbf{F}_{\mathrm{diff}}\left(\mathbf{W}^{\mathrm{lab}}_{\mathrm{Riem}}, (\partial\mathbf{W}/\partial\mathbf{r})^{\mathrm{lab}}_{\mathrm{approx}}\right)\hat{\mathbf{n}}_{ij}, \tag{3.41}$$

where $\mathbf{W}^{\mathrm{lab}}_{\mathrm{Riem}}$ is the primitive variable vector at the centroid of the interface, as seen in the lab frame (whose associated conserved variables are $\mathbf{U}^{\mathrm{lab}}_{\mathrm{Riem}}$ in Eq. 3.38). The spatial derivatives $(\partial\mathbf{W}/\partial\mathbf{r})^{\mathrm{lab}}_{\mathrm{approx}}$ correspond to our extrapolate-and-average scheme for linearly varying gradients. As with $\mathbf{W}^{\mathrm{lab}}_{\mathrm{Riem}}$, we are interested in estimates of $\partial\mathbf{W}/\partial\mathbf{r}$ at the centroid of the face. For both these quantities, only the velocity and its spatial derivatives are relevant when viscous fluxes are calculated.

3.3.3 Hessian Estimation

In analogy to the gradient calculation for Voronoi meshes discussed by Springel (2010a), here we discuss the estimates of the cell-centered Hessian matrices for each of the primitive variables W_α. To this end, let us consider a vector field $\mathbf{u}$ that varies approximately linearly with distance as $\mathbf{u} \approx \mathbf{u}_i + \mathbf{h}\,(\mathbf{r} - \mathbf{r}_i)$ near $\mathbf{r}_i$. Up to linear order, the first derivative of $\mathbf{u}$ is simply $\mathbf{h}$. The volume-average of the spatial derivatives of $\mathbf{u}$ in the vicinity of $\mathbf{r}_i$ is

$$\begin{aligned} V_i \left\langle \frac{\partial \mathbf{u}}{\partial \mathbf{r}} \right\rangle_i &= \int_{V_i} \frac{\partial \mathbf{u}}{\partial \mathbf{r}}\, \mathrm{d}V \\ &= \int_{\partial V_i} \mathbf{u}\, \mathrm{d}\mathbf{A} \\ &= \sum_{j \neq i} \int_{A_{ij}} \left[\mathbf{u}_i + \mathbf{h}(\mathbf{r} - \mathbf{r}_i)\right] \frac{\mathbf{r}_j - \mathbf{r}_i}{r_{ij}}\, \mathrm{d}A, \end{aligned} \tag{3.42}$$

where we have assumed that the linear approximation is valid up to all the neighboring mesh generating points $\mathbf{r}_j$. It is straightforward to verify that the average matrix $\langle \partial \mathbf{u}/\partial \mathbf{r} \rangle_i$ can be written as

$$\begin{aligned} \left\langle \frac{\partial \mathbf{u}}{\partial \mathbf{r}} \right\rangle_i =& \frac{1}{V_i} \sum_{j \neq i} A_{ij} \left(\frac{\mathbf{u}_i + \mathbf{u}_j}{2} \otimes \widehat{\mathbf{n}}_{ij} \right) \\ &- \frac{1}{V_i} \sum_{j \neq i} A_{ij} \left(\mathbf{h}\, \mathbf{c}_{ij} \otimes \frac{\mathbf{r}_{ij}}{r_{ij}} \right). \end{aligned} \tag{3.43}$$

Writing the vector product $(\mathbf{A}\,\mathbf{u}) \otimes \mathbf{v}$ in tensor form (where $\mathbf{A}$ is a $n \times n$ square matrix and $\mathbf{u}$ and $\mathbf{v}$ are vectors of dimension n), it is easy to prove the identity $A_{ac} u_c v_b = A_{ac} v_c u_b + \varepsilon_{bfc} \varepsilon_{fde} u_d v_e A_{ac}$, where ε_{abc} is the Levi-Civita symbol. Equivalently, going back to vector notation, we have $(\mathbf{A}\,\mathbf{u}) \otimes \mathbf{v} = (\mathbf{A}\,\mathbf{v}) \otimes \mathbf{u} + (\mathbf{u} \times \mathbf{v}) \times \mathbf{A}$, where, for simplicity, we used vector notation to denote a "cross product" between a vector and a matrix.

Therefore, the second term on the right hand side of Eq. (3.43) can be written as

$$\begin{aligned}\sum_{j\neq i} A_{ij}\left(\mathbf{h}\,\mathbf{c}_{ij}\otimes\frac{\mathbf{r}_{ij}}{r_{ij}}\right) =& \sum_{j\neq i} A_{ij}\left(\mathbf{h}\,\mathbf{r}_{ij}\otimes\frac{\mathbf{c}_{ij}}{r_{ij}}\right)\\ &+\sum_{j\neq i}\left(A_{ij}\mathbf{c}_{ij}\times\frac{\mathbf{r}_{ij}}{r_{ij}}\right)\times\mathbf{h}\quad .\end{aligned} \tag{3.44}$$

Here, the second term on the right hand side vanishes identically, because

$$\begin{aligned}\sum_{j\neq i}\left(A_{ij}\mathbf{c}_{ij}\times\frac{\mathbf{r}_{ij}}{r_{ij}}\right)\times\mathbf{h} &= \left\{\int_{\partial V_i}\left(\mathbf{r}-\frac{\mathbf{r}_i+\mathbf{r}_j}{2}\right)\times\mathrm{d}\mathbf{A}\right\}\times\mathbf{h}\\ &= \left\{\int_{V_i}\nabla\times\left(\mathbf{r}-\frac{\mathbf{r}_i+\mathbf{r}_j}{2}\right)\mathrm{d}V\right\}\times\mathbf{h}\\ &= 0\quad .\end{aligned} \tag{3.45}$$

On the other hand, the first term on the right hand side of Eq. (3.44) can be rewritten by means of the replacement $\mathbf{h}\,\mathbf{r}_{ij} = -\mathbf{h}\,(\mathbf{r}_j-\mathbf{r}_i) = \mathbf{u}_i-\mathbf{u}_j$. Finally, identifying the vector $\mathbf{u}_i$ with the gradient $\langle\nabla\phi\rangle_i$ of a scalar quantity ϕ, and the matrix $\langle\partial\mathbf{u}/\partial\mathbf{r}\rangle_i$ with the cell-averaged Hessian matrix $\langle\mathbf{H}^{\phi}\rangle_i$, Eq. (3.44) takes the form

$$\begin{aligned}\left\langle\mathbf{H}^{\phi}\right\rangle_i =& \frac{1}{V_i}\sum_{j\neq i}A_{ij}\left\{-\left(\frac{\langle\nabla\phi\rangle_i+\langle\nabla\phi\rangle_j}{2}\right)\otimes\frac{\mathbf{r}_{ij}}{r_{ij}}\right.\\ &\left.+\left(\langle\nabla\phi\rangle_j-\langle\nabla\phi\rangle_i\right)\otimes\frac{\mathbf{c}_{ij}}{r_{ij}}\right\}.\end{aligned} \tag{3.46}$$

The most noteworthy characteristic of this expression is that it is purely algebraic and explicit in nature. That is, the Hessian matrix of ϕ is simply a linear combination of the neighboring gradients in which the coefficients are predetermined quantities that depend only on the local mesh geometry. Each one of those neighboring gradients is, at the same time, a linear combination of its immediate neighbors' scalar quantities (see Eq. 21 of Springel, 2010a). Therefore, the Hessian estimate of Eq. (3.46) is a weighted linear combination of scalars from its immediate neighbors and from its neighbors' neighbors and, as such, it implicitly employs a larger stencil than the one used for the gradients.

3.3.4 Slope-Limiting the Hessians

It is well known that higher-order reconstruction schemes are prone to produce spurious oscillations in the vicinity of steep gradients, unless this is prevented by appropriate *slope limiter methods* (Toro, 2009). Similarly, potential irregularities in the second derivative fields can lead to spurious oscillations and unphysical values of the viscous stress tensor at the cell boundaries. These unphysical values would be self-regulated by negative feedback, since large viscous fluxes diffuse angular momentum efficiently, thus eliminating spurious fluctuations as soon as these arise. However, we have found in our experiments that the viscous kick predictor step (Equation 3.26) can be affected by noise values of the *second* derivatives, thus producing unphysical velocities before the MUSCL-Hancock step is carried out. To alleviate this problem, we have included a "safety mechanism" that consists of "limiting" the coefficients of the Hessian matrix in a way very much similar to the slope-limitation of the gradients. This is done by correcting

$$\overline{\left\langle \mathbf{H}^{\phi} \right\rangle_i} = \mathbf{A}_i \left\langle \mathbf{H}^{\phi} \right\rangle_i \tag{3.47}$$

where $\mathbf{A}_i$ is a diagonal matrix of slope-limiting coefficients.

In general, the second derivatives of the velocity field should vary rather gently in space. Indeed, for regions of smooth flow, the gradients will also be smooth and therefore the Hessians will be small and smooth. For regions of shocked flow, the gradients will be slope-limited and thus the Hessians should be small regardless. Therefore, the correction proposed here should only affect very pathological regions of the flow and thus behaves more as a "switch" rather than a technique for numerical stability. It is worth pointing out that in almost all our tests (those with reasonably smooth initial conditions) we can run without such switch, with unchanged results.

3.3.5 Time integration and time-step criterion

Because of the more complex mathematical properties of the NS equations compared with the Euler equations, obtaining a rigorous analytic expression analogous to the CFL stability criterion for the allowed time step size is not possible. However, MacCormack & Baldwin (1975) obtained an approximate semi-empirical stability criterion when advective, viscous and heat diffusion terms are considered. When there is no heat flux, the time-step criterion can be written as (e.g. Kundu & Cohen, 2008)

$$\Delta t \leq \frac{\sigma \Delta t_{\rm CFL}}{1 + 2/\{\mathrm{Re}\}_i}, \tag{3.48}$$

where $\Delta t_{\rm CFL}$ is the standard CFL-criterion time-step except for the Courant-Friedrichs-Levy coefficient, which is absorbed into a "safety factor" σ (usually ≈ 0.9). In Eq. (3.48), the *cell Reynolds number* $\{\mathrm{Re}\}_i$ is

$$\{\mathrm{Re}\}_i = \frac{\rho |\mathbf{v}_i| R_i}{\eta}, \tag{3.49}$$

where $\mathbf{v}_i$ is the physical velocity of the flow evaluated at the i-th mesh point, and R_i is the effective radius of the cell, calculated as $R_i = (3V_i/4\pi)^{1/3}$ from the volume of a cell (or as $R_i = (A_i/\pi)^{1/2}$ from the area in 2D). Similar approaches to derive an appropriate NS time-step have also been described by Mavriplis & Jameson (1990) and Coirier & Powell (1996).

The numerical integration scheme we employ is time unsplit, that is, advective and diffusive fluxes are applied simultaneously during each hydrodynamic time-step and not sequentially (Eq. 3.25). The prediction stage, on the other hand, is operator-split, since the advective and diffusive terms are computed almost independently of each other. This is in part due to the nature of the standard one-dimensional Riemann problem, whose

solutions – strictly speaking – are only valid for the hyperbolic Euler problem, but are not solutions to the full NS equations with their additional parabolic terms.

3.4 Numerical Test Results

To test the performance of AREPO when our new treatment of viscous fluxes is included, we have carried out a number of test simulations for physical situations with known analytic or quantitative solutions. Usually, the problems with known exact solutions are either of self-similar type or have symmetries that make the non-linear term proportional to $(\mathbf{v} \cdot \nabla)\mathbf{v}$ vanish identically. Owing to these limitations, numerical simulations of situations with experimentally well-established behavior, such as flow past a circular cylinder, have become common-place in testing the performance of NS codes. We will therefore also carry out such qualitative benchmarks, besides looking at a few simple problems with analytic solutions.

3.4.1 Diffusion of a Vortex Sheet

A simple problem of laminar flow in the presence of viscosity is given by the vortex sheet diffusion test. In this problem, the initial velocity field at $t = 0$ is given by $\mathbf{v} = (u, 0, 0)$ with $u = 1$ for $y > 0$ and $u = -1$ for $y < 0$. Because of the symmetry of the problem, the NS equations reduce to a 1D diffusion equation

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial^2 y}, \tag{3.50}$$

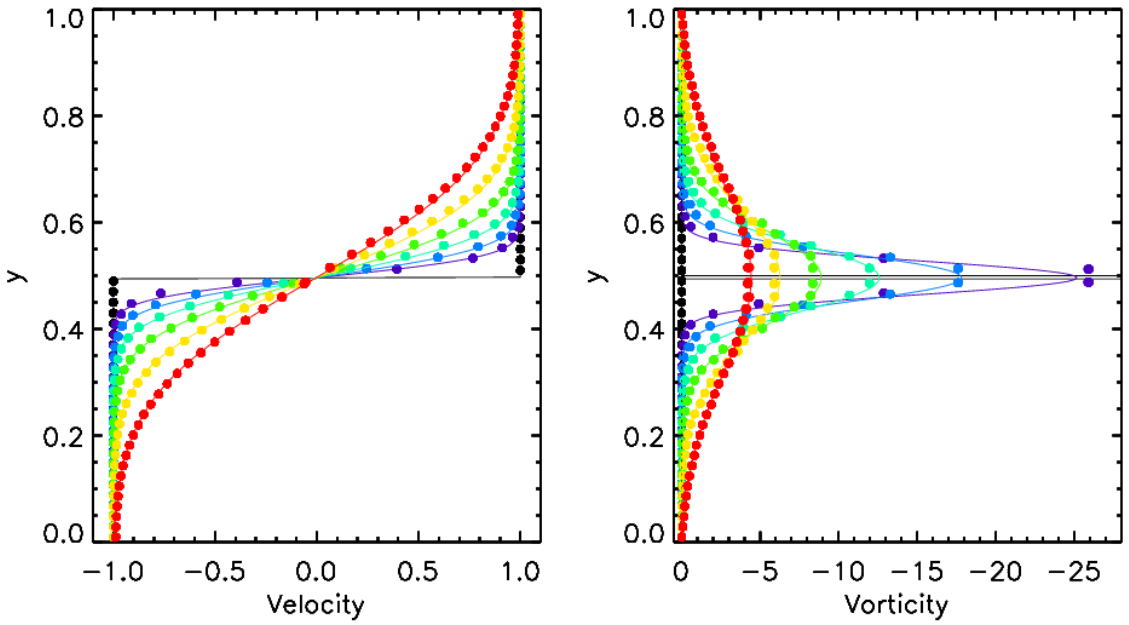
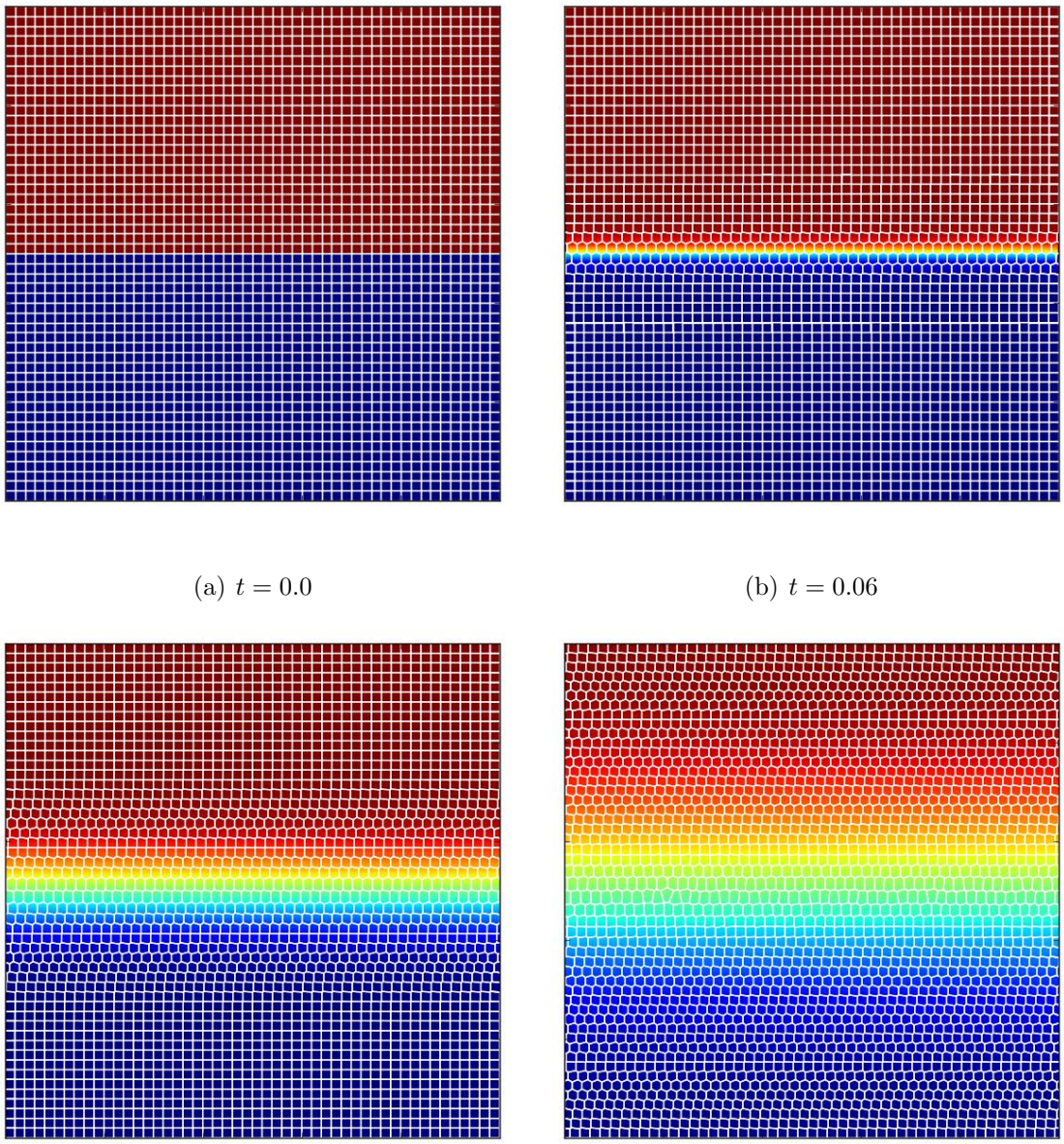


Figure 3.5.— Diffusion of a vortex sheet. The two panels show the velocity u along the x-axis (*left panel*), and the vorticity (*right panel*), at times $t = 0$, 0.1, 0.2, 0.4, 0.8, 1.6 and 3.2 (from black to red), for a dynamic viscosity coefficient $\mu = \nu\rho = 0.005$. The solid lines are given by the analytic solution described by Eqs. (3.51), while the solid circles are *all* 2500 cell-centered velocity and vorticity values of the initially Cartesian 50×50 mesh. Note that the simulation is started with a sharp discontinuity in velocity and thus the δ-function vorticity field is initially unresolved. If the mesh would remain exactly Cartesian, the diffusion of vorticity would actually be suppressed in this case. Nevertheless, the small asymmetries introduced by the moving mesh trigger the diffusion regardless of the initially unresolved setup, and the time-dependent numerical result closely follows the expected exact solution.



(a) $t = 0.0$ (b) $t = 0.06$

(c) $t = 0.6$ (d) $t = 1.8$

Figure 3.6.— Time evolution of the mesh geometry and the velocity field for a diffusing vortex sheet test. As the vorticity spreads from the center of the domain to the upper and lower boundaries, the mesh adapts to the continuous change in velocity until its original Cartesian structure disappears entirely. The color table (from blue to red) corresponds to the range between $u = -1.0$ and $u = 1.0$ in linear scale.

with solution (e.g. Kundu & Cohen, 2008)

$$u = \mathrm{erf}\left[\frac{y}{2\sqrt{\nu t}}\right] \qquad \omega = \frac{\partial u}{\partial y} = \frac{e^{-y^2/4\nu t}}{\sqrt{\pi \nu t}}. \tag{3.51}$$

In Figure 3.5, we show the time evolution we obtain for a two-dimensional simulation domain with initially uniform pressure and density ($\rho = P = 1$), and with a velocity field given by $\mathbf{v} = (\mathrm{sgn}(y), 0, 0)$. The mesh generating points were distributed regularly at the initial time to produce a Cartesian mesh. As the system evolves, the velocity and the vorticity fields as a function of time and vertical coordinate y follow the exact solution remarkably well. It is worth pointing out that the initial singularity in the vorticity field is unresolved numerically (and thus appears as being uniformly zero throughout the domain), since the system is started with an exact sharp discontinuity. Static, perfectly aligned meshes with slope limitation techniques will typically maintain this unresolved vorticity and thus no diffusion will proceed unless some numerical perturbations are seeded that break the mesh alignment of the initial state (a common way to overcome this difficulty is to start the system according to Eq. (3.51) at $t > 0$ such that there is initial vorticity). However, the moving mesh of AREPO "sees" a non-zero velocity gradient as soon as the upper and lower halves of the domain become unaligned with respect to each other. This happens because, as soon as a cell shifts its position, the number of its neighbors that have a drastically different velocity increases and so does the "statistical weight" of the discontinuity. At this point, the slope-limiting technique, which had ignored the discontinuity in the perfectly aligned mesh, now identifies the local variation as "real" and the vorticity field is "detected".

Fig. 3.6 shows the corresponding two-dimensional velocity field of the diffusing vortex sheet test at four different times, together with the geometry of the underlying

Voronoi mesh. The mesh geometry nicely shows how the cells transform from a Cartesian configuration to an unstructured mesh, while the velocity field evolves from a piece-wise constant state with a central discontinuity to a smoothly varying shear flow due to the effects of viscosity.

3.4.2 Diffusion of a Viscous Vortex

The two-dimensional circular velocity distribution corresponding to an irrotational vortex of circulation Γ is

$$v_\theta = \frac{\Gamma}{2\pi R}, \tag{3.52}$$

where the vorticity $\omega = |\nabla \times \mathbf{v}| = (1/R)\partial(R\, v_\theta)/\partial R$ is zero everywhere except at the origin ($\omega = \delta(R)$, i.e. a vortex line). In a viscous fluid, this velocity profile has to be sustained by a point source of vorticity at the origin (e.g. an infinitely thin rotating cylinder) otherwise the vortex line will decay in a similar way as the vortex sheet in the previous example. If the velocity at the origin is set impulsively to zero, the subsequent evolution of the azimuthal velocity is given by

$$v_\theta(R,t) = \frac{\Gamma}{2\pi R}\left[1 - e^{-R^2/4\nu t}\right] \quad , \tag{3.53}$$

while the vorticity $\omega = \left[\nabla \times (v_\theta \hat{\boldsymbol{\theta}})\right] \cdot \hat{\mathbf{z}}$ evolves as

$$\omega = -\frac{\Gamma}{4\pi\nu t} e^{-R^2/4\nu t} \tag{3.54}$$

and the Laplacian of the velocity field is

$$|\nabla^2 \mathbf{v}| = \frac{\Gamma}{2\pi}\frac{R}{(2\nu t)^2} e^{-R^2/4\nu t}\hat{\boldsymbol{\theta}} \quad . \tag{3.55}$$

Because of its geometry, this problem is significantly more challenging than the vortex sheet test considered above and cannot be impulsively started at precisely $t = 0$. Besides

the initial singularity in the vorticity field, the velocity field is divergent as we approach the origin. In addition, it is not possible to capture the azimuthal velocity field when the distance from the origin is comparable to the grid resolution. At the same time, the azimuthal velocity field is challenging for the boundary conditions, because the problem is self-similar in nature and therefore natural boundaries do not exist. These problems did not exist for the vortex-sheet problem, which is of one-dimensional nature. Nevertheless, evolving the system from an initial time $t > 0$ minimizes most of these complications. In addition, we extend the computational domain far beyond the region of interest, such that boundaries become essentially irrelevant during the timespan of the numerical solution.

We setup a Cartesian mesh (100×100) with an imposed initial velocity profile of

$$v_{\theta,0} = \frac{\Gamma}{2\pi R}\left[1 - \exp\left(-\frac{R^2}{4\nu t_0}\right)\right] \qquad \text{with } \nu = \frac{\mu}{\rho}, \tag{3.56}$$

corresponding to a Gaussian vortex that we center in the middle of the domain, which extends over the range $[0, 40] \times [0, 40]$, and thus accommodates a radial range from $R = 0$ to $R = 20$. The adopted physical parameters are $t_0 = 10$, $\mu = 0.08$, $\Gamma = 1.0$, and the initial density field is constant with $\rho = 1$. The pressure field, however, is not uniform because the fluid is not started from rest. We obtain the correct pressure profile from the radial component of the equation of motion:

$$-\frac{v_\theta^2}{R} = -\frac{1}{\rho}\frac{\mathrm{d}P}{\mathrm{d}R},$$

and thus the initial pressure profile is

$$P_{\text{init}} = P_0 - \frac{\Gamma^2\rho}{4\pi^2}\left\{\frac{1}{2R^2}e^{-R^2/(2\nu t_0)}\left[e^{R^2/(4\nu t_0)} - 1\right]^2 + \frac{1}{4\nu t_0}\left(\mathrm{Ei}\left(-\frac{R^2}{2\nu t_0}\right) - \mathrm{Ei}\left(-\frac{R^2}{4\nu t_0}\right)\right)\right\} \quad ,$$

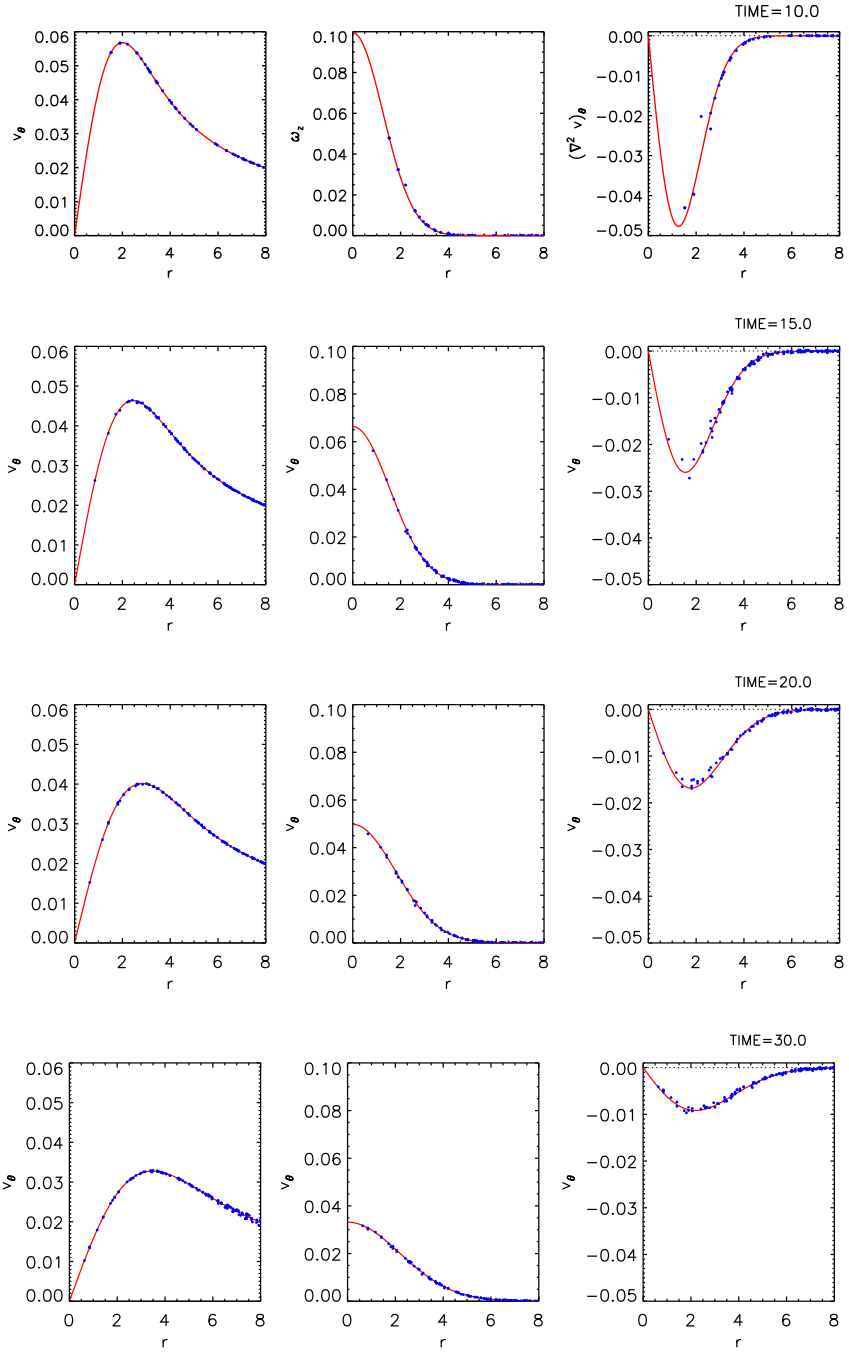


Figure 3.7.— Time evolution of a diffusing Gaussian vortex. For each time (as labeled), we show the azimuthal velocity profile $v_\theta(R)$, the vorticity profile $\omega(R)$ and the Laplacian profile $\nabla^2 v_\theta$, as computed by AREPO (blue points; only a random 10% of the total shown) and compare it to the corresponding analytic expressions (solid red lines).

where P_0 is an integration constant. The precise value of P_0 is irrelevant for the similarity solution presented here, because it is obtained for incompressible flow. In our numerical experiments (which are compressible), we set P_0 such that $P = 1$ at $R = 0$.

Fig. 3.7 shows the time evolution of the velocity field, the vorticity field and the Laplacian field for a Gaussian vortex started on an initially Cartesian mesh. We find not only that the velocity evolves as expected based on the similarity solution, but the first and second derivatives also show excellent agreement with the analytic expectations. These results validate both the space- and time-accuracy of our viscous integration scheme, as well as the accuracy with which the second derivatives are estimated.

A similar test is the triangular vortex of Gresho & Chan (1990). This vortex is described by the azimuthal velocity profile

$$v_\phi(R) = \begin{cases} 5R & 0 \leq R < 0.2 \\ 2 - 5R & 0.2 \leq R < 0.4 \\ 0 & 0.4 \leq R \end{cases} \tag{3.57}$$

and corresponds to a steady-state solution of the Euler equations when a suitable pressure profile is provided (Liska & Wendroff, 2003; see also Springel, 2010a for an implementation in AREPO). In the presence of explicit viscosity, the evolution of the Gresho vortex becomes time-dependent, with the velocity field evolving in a qualitatively similar way to the Gaussian vortex (see Figure 3.8).

We use this setup to measure the convergence rate of AREPO when viscous fluxes are included. Contrary to the Gaussian vortex, this problem has no analytic solution as a function of time, since a similarity solution cannot be obtained. We use a high-resolution run (1280^2) as an "exact" or reference solution, and compare that to a set

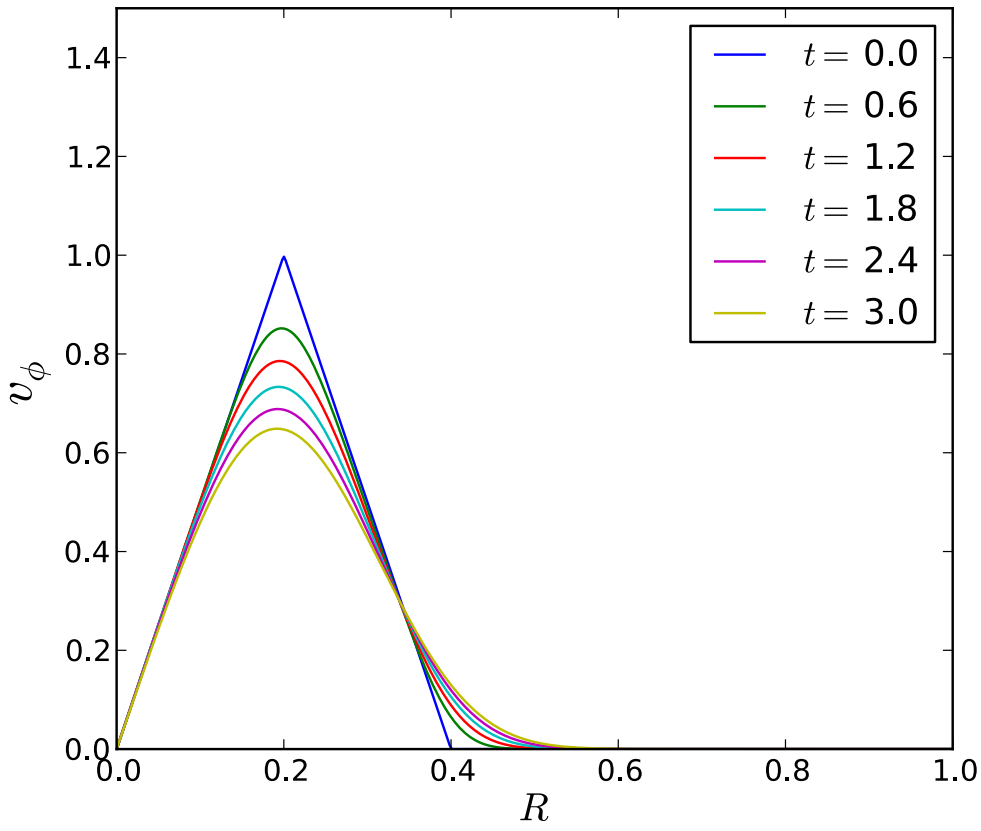


Figure 3.8.— Azimuthally averaged time evolution of a Gresho vortex with viscosity coefficient $\mu = 10^{-3}$ in a high-resolution run of 1280^2 cells.

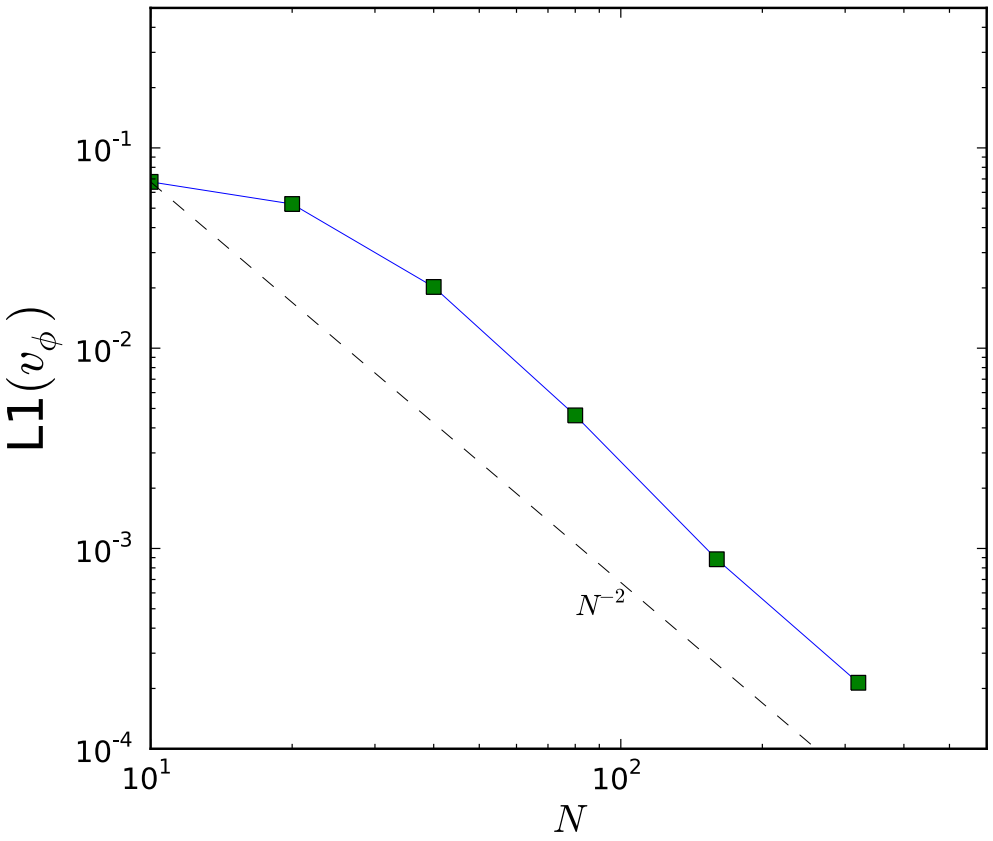


Figure 3.9.— L^1-error norm for the viscous Gresho vortex at time $t = 3.0$ (internal units). Six different runs, of resolutions $N \times N$ with $N = 10, 20, 40, 80, 160$ and 320, show an error decline proportional to N^{-2}, indicative of second-order accuracy. The error residuals are measured respect to a high-resolution run consisting of 1280^2 resolution elements.

of low-resolution runs by means of the L^1-error norm. The decline of the error residuals of runs at 10^2, 20^2, 40^2, 80^2, 160^2 and 320^2 (see Figure 3.9) shows that the time-dependent solution converges toward the high-resolution run at the expected second-order rate.

Note that Springel (2010a) finds a convergence rate shallower than second-order for this same problem in the absence of explicit viscosity. The discrepancy with the theoretical second-order accuracy is presumably attributable to the discontinuous velocity gradient imposed by the setup. Since the inclusion of a diffusive term smooths out discontinuities in the velocity field, the viscous solution can achieve full second-order accuracy.

3.4.3 Plane Poiseuille and Couette Flows

Next, we consider impulsively-started plane Poiseuille and Couette flows where a fluid between two parallel plates is initially at rest, and then, suddenly, either pressure gradients or plate motions are applied. The time-dependent solution has the form $\mathbf{v} = (u(y,t), 0, 0)$, where the horizontal velocity can be decomposed into steady and time-dependent parts, $u(y,t) = u_0(y) + \widetilde{u}(y,t)$. In the presence of a pressure gradient and an upper plate moving at constant speed U, the steady state solution is the well-known expression

$$u_0(y) = \frac{yU}{b} - \frac{y}{2\mu}\frac{\mathrm{d}p}{\mathrm{d}x}(b-y)\,, \tag{3.58}$$

for which the special cases $U = 0$ and $\mathrm{d}p/\mathrm{d}x = 0$ are commonly known as plane Poiseuille flow and plane Couette flow, respectively.

The time dependent component $\widetilde{u}(y,t)$ is a solution of Eq. (3.50), subject to the initial condition $\widetilde{u}(y,0) = -u_0(y)$ and the boundary conditions $\widetilde{u} = 0$ at $y = 0$ and $y = b$.

By separation of variables, the general solution is (e.g Graebel, 2007)

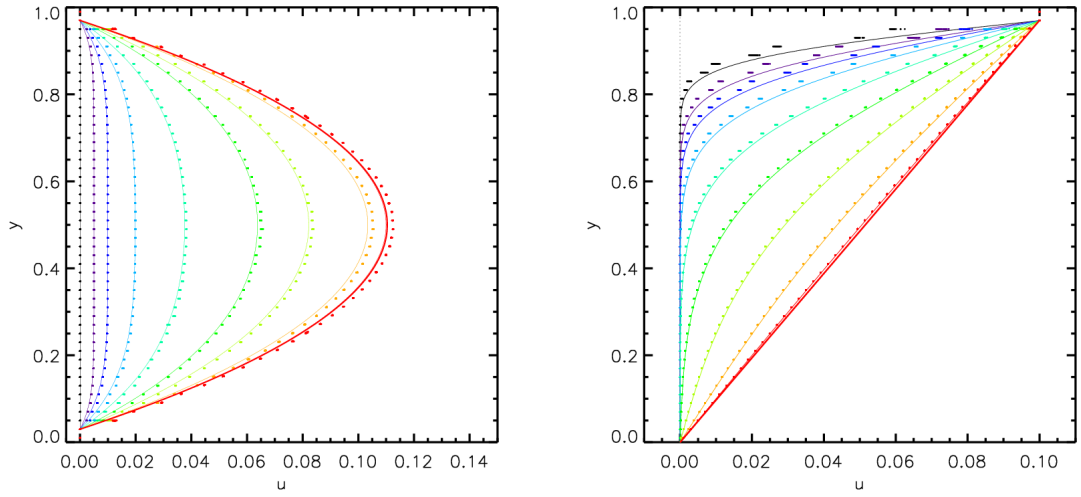
$$\widetilde{u}(y,t) = \sum_{n=1}^{\infty} A_n e^{-n^2\pi^2\nu t/b^2} \sin\frac{n\pi y}{b}, \tag{3.59}$$

where the coefficients A_n are determined by the initial condition

$$A_n = -\frac{\int_0^b u_0(y)\sin\frac{n\pi y}{b}\mathrm{d}y}{\int_0^b \sin^2\frac{n\pi y}{b}\mathrm{d}y} \tag{3.60}$$

$$= -\frac{2U(-1)^n}{n\pi} - \frac{2}{b\mu}\frac{\mathrm{d}p}{\mathrm{d}x}\left(\frac{b}{n\pi}\right)^3\left[1-(-1)^n\right] \quad . \tag{3.61}$$

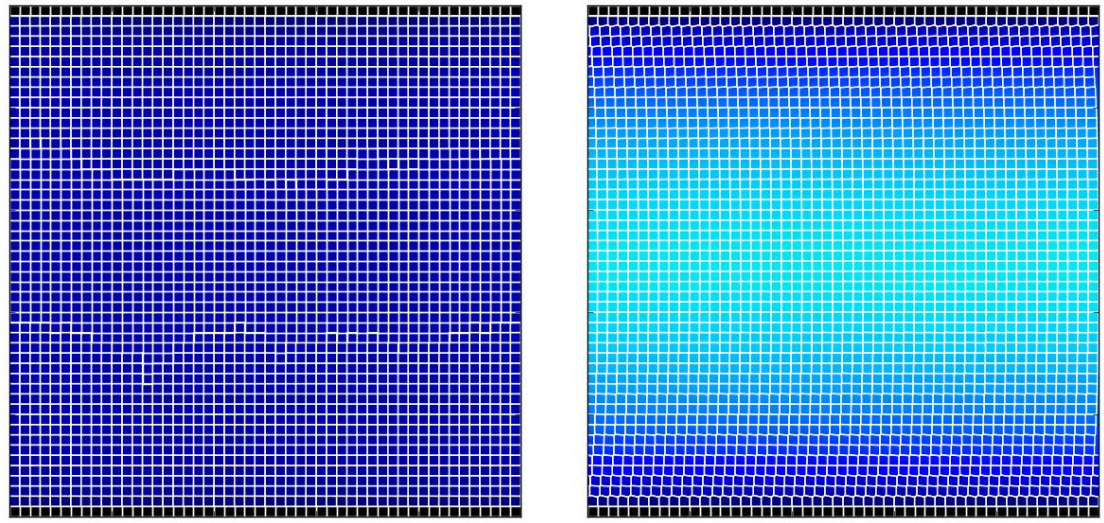
The numerical setup for this problem is straightforward. We produce a Cartesian mesh in the range $[0,1]\times[0,1]$ with a resolution of 50×50. The fluid is originally at rest and its density and pressure are given by $\rho = P = 1$. The equation of state is that of an ideal gas with adiabatic index 5/3. To represent the plates, the uppermost and lowermost rows of cells are replaced by "solid boundaries" at which the no-slip condition is enforced, i.e. $v_x = v_y = 0$ (see Fig. 3.12). Moving solid boundaries are straightforward to implement with a Voronoi tessellation mesh. A solid surface can be constructed as a series of mesh-generating point pairs, one on each side of the surface, such that the common interface – equidistant to both points – defines the boundary locally (see Serrano & Español, 2001 and Springel, 2010a). The Voronoi cell on the side of the "solid" object can regarded as "a ghost cell within the domain". That is, this cell is part of the domain discretization process and is tessellated/updated as any other normal gas cell. However, when solving the Riemann problem at the local interface between a "solid" cell and a real gas cell, boundary conditions are imposed in the same way as boundary conditions on the outer box are imposed. For perfectly reflecting boundaries, the normal component of the velocity is reflected in the "solid side" or "outside region" of the interface. For non-slip boundaries, the entire velocity vector is reflected, such that



(a) Plane Poiseuille flow

(b) Plane Couette flow

Figure 3.10.— Impulsively started plane Poiseuille and Couette flows as a function of time. a) Time evolution of the horizontal velocity profile versus vertical distance. Solid curves represent the analytic solutions of Eqs. (3.58) to (3.60) for $U = 0$ and $\mathrm{d}p/\mathrm{d}x = -0.05$, at ten different times (time increasing from black to red). The data points correspond to all the cell-centered values of velocity along x for a 50×50 simulation started from rest. b) Time evolution of the horizontal velocity profile versus vertical distance. Solid curves represent the analytic solutions in Eqs. (3.58) to (3.60) for $U = 0.1$ and $\mathrm{d}p/\mathrm{d}x = 0$ at eight different times (time increasing from black to red). The data points correspond to all the cell-centered values of velocity along x for a 50×50 numerical simulation started from rest.



(a) $t = 0.1$ (b) $t = 0.9$

(c) $t = 2.5$ (d) $t = 5.7$

Figure 3.11.— Time evolution of the mesh geometry and the velocity for flow between parallel plates. The horizontal velocity field u for plane Poiseuille flow is rendered at four different times. The evolution of the velocity field (see Fig 3.10(a)) is accompanied by the evolution of the mesh from an initially Cartesian set up (top-left panel) to a fully unstructured grid by the time the flow has reached steady state (bottom-right panel). The (linear) color scale ranges from blue ($u = 0$) to red ($u = 0.12$).

the velocity at the interface is zero (Figure 3.12).

We run two different test problems. For the first one, both plates remain at rest and an external gradient of $\mathrm{d}p/\mathrm{d}x = -0.05$ is imposed. For the second test, the bottom plate is at rest and the upper plate moves at a constant speed of $U = 0.1$. In both test simulations, the dynamic viscosity coefficient has been set to $\mu = 0.05$. In Figure 3.10, we show the time evolution of the horizontal velocity profile both for the plane Poiseuille and Couette flows. In both cases, the numerical results match the analytic expectations very well. In Figures3.11 we also show maps of the velocity profile and the mesh geometry at different times for the Poiseuille case. The grid evolution shows how the Cartesian structure is progressively lost, but that the dynamic Voronoi mesh of AREPO successfully avoids any mesh-tangling effects.

3.4.4 Time-Dependent Circular Couette Flow

We now turn to a more challenging problem, which highlights the ability of our scheme to deal with geometrically complex boundary conditions. For purely azimuthal motion, the NS equations in the radial and tangential directions are

$$-\frac{v_\theta^2}{R} = -\frac{1}{\rho}\frac{\mathrm{d}P}{\mathrm{d}R} \tag{3.62a}$$

$$\rho\frac{\partial v_\theta}{\partial t} = \mu\frac{\mathrm{d}}{\mathrm{d}R}\left[\frac{1}{R}\frac{\mathrm{d}}{\mathrm{d}R}(Rv_\theta)\right] \quad . \tag{3.62b}$$

The exact solution of steady flow (i.e. $\partial v_\theta/\partial t = 0$) between concentric cylinders with boundary conditions $v_\theta = \Omega_1 R_1$ at $R = R_1$, and $v_\theta = \Omega_2 R_2$ at $R = R_2$ is given by (e.g Kundu & Cohen, 2008)

$$v_{\theta,0}(R) = \frac{\left(\Omega_2 R_2^2 - \Omega_1 R_1^2\right) R^2 - \left(\Omega_2 - \Omega_1\right) R_2^2 R_1^2}{R(R_2^2 - R_1^2)} \quad , \tag{3.63}$$

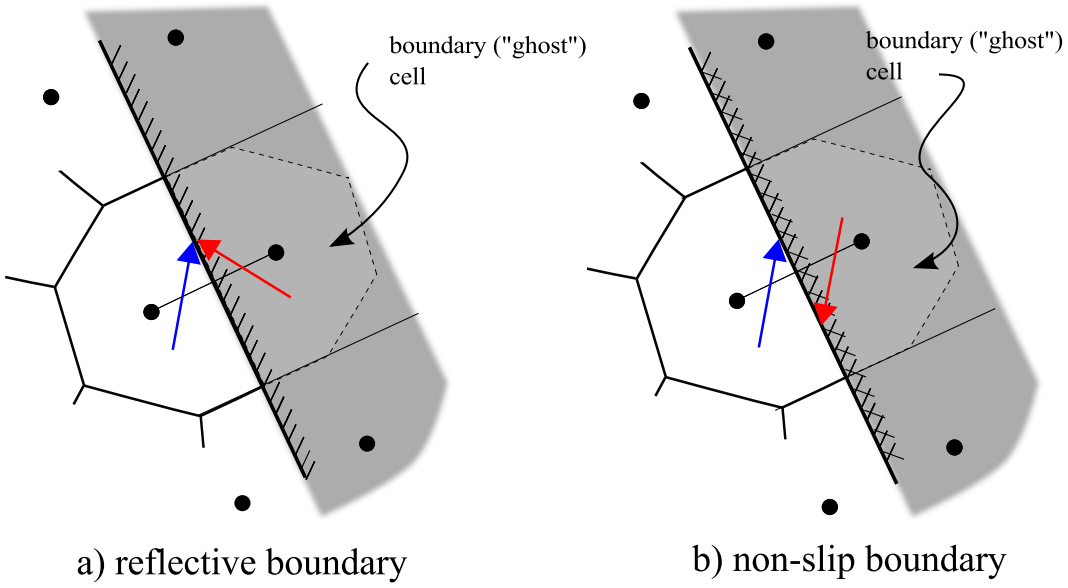


Figure 3.12.— Schematic representation of (a) reflective and (b) non-slip boundaries within the computational domain. After the spatial and temporal extrapolation steps in the MUSCL-Hancock method (panel e) in Figure 3.1), the Riemann problem is solved as elsewhere in the domain but with the boundary-side cell mimicking the gas side with either one velocity component – the normal one – reversed (reflection) or all three (non-slip).

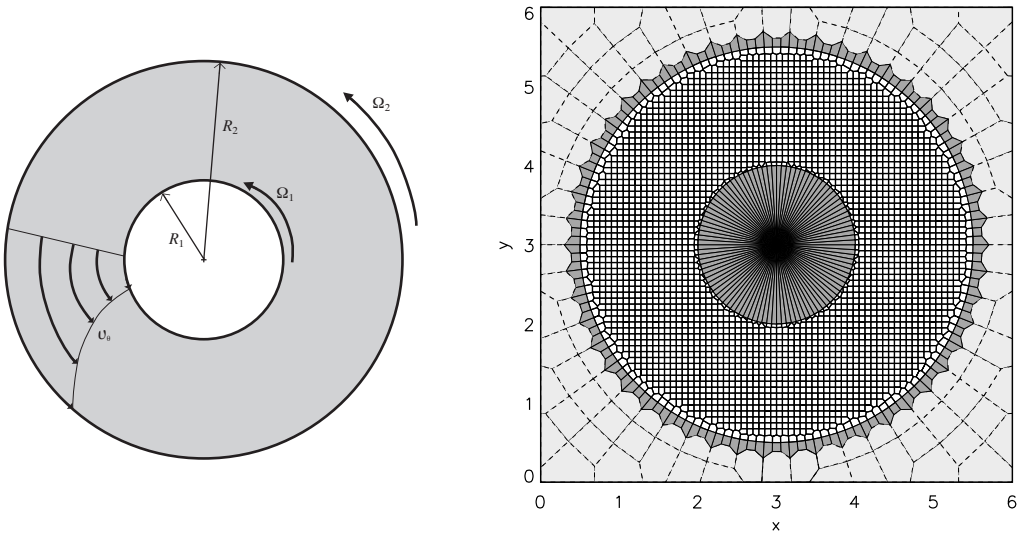


Figure 3.13.— Geometry of the circular Couette flow. The left hand panel shows a schematic view of the two-dimensional problem. The right hand panel displays the actual initial mesh used in AREPO in a setup where we start the problem impulsively from rest. Each cylindrical boundary (at radii R_1 and R_2) is generated by two layers of cells, one side representing the solid cylinder and the other representing the fluid. These layers of cells are positioned along circles. The remainder of the fluid cells, originally at rest, are distributed like a Cartesian grid. The cells outside the outer cylinder are "auxiliary cells" and are only included to fill the computational domain, but do not exert any influence on the fluid inside the cylinders. The motion of the cylinders is prescribed to remain constant (with angular velocities Ω_1 and Ω_2), and thus represents a source of kinetic energy. The motion of the fluid in between the cylinders is induced by means of the no-slip boundary condition at the contact surface, and the momentum that is transported in the radial direction through the shear viscosity.

where R_i and Ω_i $(i = 1, 2)$ are the radii and angular velocities of the respective cylinders.

The impulsively-started version of this problem can be solved analytically by separation of variables (see Tranter, 1968; Graebel, 2007). The full solution can thus be written as $v_\theta(R,t) = v_{\theta,0} + \tilde{v}_\theta(R,t)$, where the time-dependent part has the form

$$\tilde{v}_\theta(R,t) = \sum \left\{ C_2 J_1(nR) + C_2 Y_1(nR) \right\} e^{-\nu n^2 t},$$

and J_1 and Y_1 are Bessel functions of the first and second kind, respectively. This time-dependent component is subject to the boundary conditions $\tilde{v}_\theta(R,t) = 0$ at R_1 and R_2, thus allowing us to eliminate C_2:

$$\tilde{v}_\theta(R,t) = \sum_{s=1}^{\infty} \frac{A_s}{Y_1(n_s R_1)} B_1(n_s R) e^{-\nu n_s^2 t},$$

where the n_s are the roots of the equation $B_1(nR) = 0$ with $B_1(nR) \equiv J_1(nR)Y_1(nR_1) - Y_1(nR)J_1(nR_1)$. Finally, the coefficients A_s are determined by imposing the initial condition $\tilde{v}_\theta = -v_{\theta,0}$ at $t = 0$. To solve for each coefficient independently, the steady state solution must be written in terms of a series expansion of $v_{\theta,0}$ in the basis functions $B_1(n_s R)$. After some algebraic manipulations, we obtain

$$v_{\theta,0}(R) = \pi \Omega_2 R_2 \sum_{s=1}^{\infty} \frac{J_1(n_s R_2)}{J_1^2(n_s R_1) - J_1^2(n_s R_2)} B_1(n_s R) \times \left[J_1(n_s R_1) - J_1(n_s R_2) \frac{\Omega_1}{\Omega_2} \frac{R_1}{R_2} \right],$$

and therefore the time-dependent component is given by

$$\tilde{v}_\theta(R,t) = -\frac{\pi}{R} \sum_{s=1}^{\infty} \frac{J_1(n_s R_2)}{J_1^2(n_s R_1) - J_1^2(n_s R_2)} B_1(n_s R) e^{-\nu n_s^2 t} \times \Big[\Omega_2 R_2 J_1(n_s R_1) - J_1(n_s R_2) \Omega_1 R_1 \Big] \quad .$$

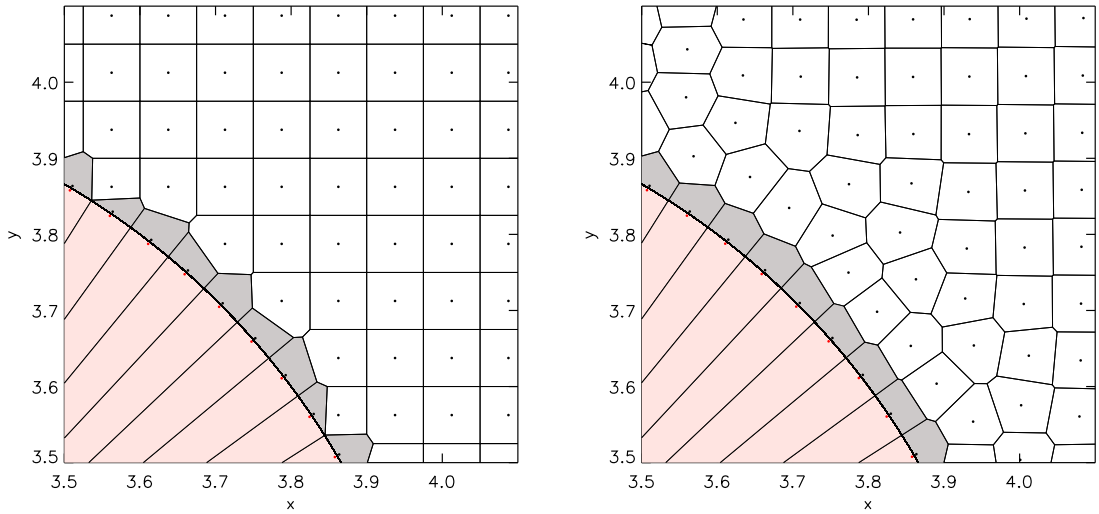


Figure 3.14.— A zoom showing the detailed mesh geometry around the inner boundary of the circular Couette flow, at two different times. The left hand panel shows a close-up view of the right panel of Fig. 3.13. The Voronoi faces that make up the cylindrical boundary are created by close pairs of points, which either lie inside the solid cylinder (red) or on the fluid side (black). The gray cells define the contact region of the fluid domain with the cylinder; here the no-slip boundary conditions are imposed. An analogous geometry applies for the outer cylinder. The panel on the right hand side shows the same region of the computational domain at a slightly later time, when the mesh filling the fluid region has started to react to the motion of the cylinder.

Collecting these results, the complete expression for the time-dependent angular velocity profile is

$$\Omega(R,t) = \frac{v_\theta}{R} - \frac{\pi}{R}\sum_{s=1}^{\infty} \frac{J_1(n_s R_2)}{J_1^2(n_s R_1) - J_1^2(n_s R_2)} B_1(n_s R) e^{-\nu n_s^2 t} \times \Big[\, \Omega_2 R_2 J_1(n_s R_1) - J_1(n_s R_2)\Omega_1 R_1 \Big] \quad . \tag{3.64}$$

We realize the moving boundary conditions in the present case through special Voronoi-cells with prescribed motion and boundary conditions, as described in Springel (2010a). In the present case, we use two sets of mesh-generating points, each one consisting of a series of outside-inside pairs located on either side of the boundary and running parallel to it, so that two circular boundaries of radii R_1 and R_2 are defined which can be made to rotate at angular frequencies Ω_1 and Ω_2, respectively. Note that the only significant technical difference between this problem and the preceding examples is the way the boundary cells are prescribed to move; the rest of the numerical scheme remains unaltered.

Figure 3.13 illustrates the geometry of the circular Couette flow, and our realization of a suitable mesh in AREPO. Since the equations of motion are always solved in the moving frame of the interfaces, there is no practical difference between stationary and moving boundaries when they are constructed as a part of the mesh. Figure 3.14 shows an enlargement of the mesh at the boundary corresponding to the inner cylinder, which is represented by a set of Voronoi faces that follow a circular path. Each one of these Voronoi faces is defined by two mesh generating points located on either side of the face, one of them outside the cylinder on the fluid side, the other inside the cylinder on the side that does not contain fluid. The right-hand panel of Fig. 3.14 shows the same region again, but at a slightly later time. This gives a sense of how the initial Cartesian mesh

between the cylinders reacts to the fluid motion. Since the latter is azimuthal, the mesh eventually develops an axial geometry, independent of the initially Cartesian setup.

Our numerical experiment for this setup has the following parameters. The initial mesh as described in Figs. 3.13 and 3.14 contains 3,254 points, out of which 2,644 are regular fluid cells, 250 are boundary fluid cells, 250 are solid boundary cells and 110 are unused auxiliary cells that are only put in to fill up the total mesh area to an enclosing rectangular shape, as presently required by AREPO. The radial distance between the cylinders is spanned by 20 cells. The physical parameters of the Couette flow are $R_1 = 1$, $R_2 = 2.5$, $\Omega_1 = 0.5$, and $\Omega_2 = 0.1$, with a dynamic viscosity coefficient set to $\mu = 0.005$. In addition, since the flow is started from rest, the pressure and density are taken to be uniform with values $\rho = P = 1$. Figure 3.15 shows the time evolution of the angular velocity profile as it asymptotically converges to the steady state solution. The agreement of the numerical data points with the exact analytic solution (Eq. 3.64) is exceptional at all times.

Finally, we show in Fig. 3.16 the mesh geometry at the end of the calculation. Even though we have started the calculation with an initially Cartesian mesh, the memory of this geometry is lost during the calculation, and the mesh dynamically adapts to the azimuthal flow structure present in this problem. The transition from a Cartesian grid towards a cylindrical-like mesh can also be seen in the output sequence of the simulation shown in Fig. 3.15, where the values of the radial position of the cells start to segregate into a set of radial "bins". The number of these radial clusters corresponds to the average number of cells along the radial direction.

It is interesting to comment on the scatter of points – especially at the beginning of

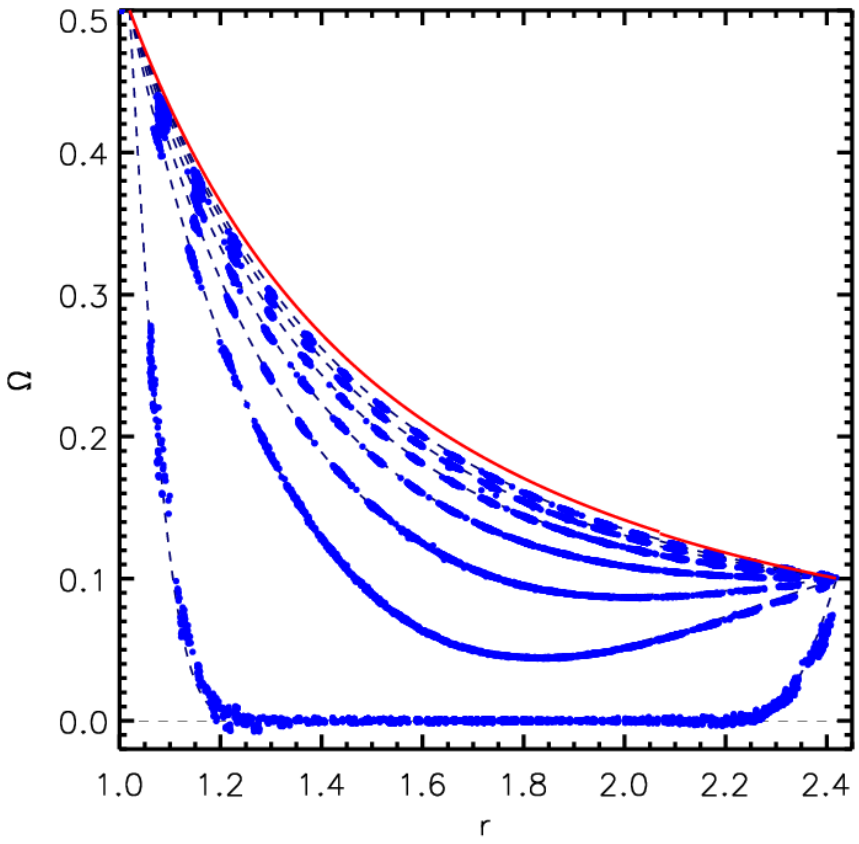


Figure 3.15.— Angular velocity profiles at different times for an impulsively started Taylor-Couette flow. For seven snapshots at times $t = 0.5$, 20, 40, 60, 80, 100, and 120 we show the cell-centered values of Ω, which are plotted as filled blue dots for all fluid cells in the calculation. No binning or averaging has been performed. The clustering of cell-center points as the system evolves is simply a consequence of the mesh adopting an axial symmetry in an adaptive fashion. The dashed lines give the time-dependent analytic solution of Eq. (3.64) at the corresponding times. The numerical results are almost indistinguishable from the exact solution. The red curve depicts the steady state solution to which the time-dependent solution eventually converges.

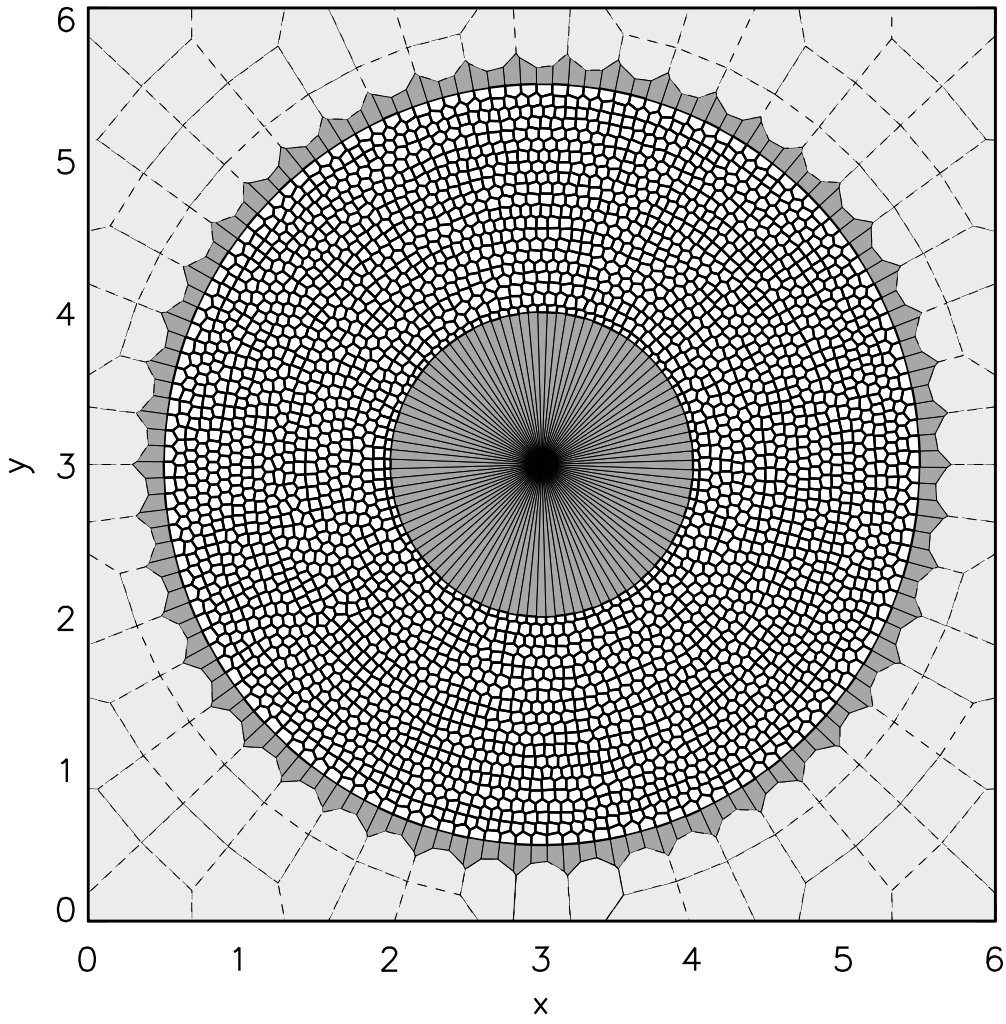


Figure 3.16.— Mesh geometry for the circular Couette flow towards the end of the numerical integration. Even though we have started the calculation with an initially Cartesian mesh, this structure is quickly lost in favor of an on average azimuthal mesh geometry.

the simulation – as seen in the angular velocity profile of Figure 3.15. This is a reflection of the challenging initial mesh geometry. Although high-order schemes – fifth or sixth order – are not sensitive to the compliance of the mesh geometry with the flow, second order schemes are. In this particular case, an axially symmetric mesh geometry would be more suitable due to the characteristics of the flow. However, the main point of this test is to show how the mesh responds to the evolution of the problem, achieving rough axisymmetry despite the unfavorable initial setup.

As discussed by Springel (2010a), our moving Voronoi mesh technique needs a "quality control" to keep cells sufficiently regular in order to avoid large errors in the spatial reconstruction. However, this modification of the mesh motion comes at a price: imagine a very strong compression along one direction (e.g. due to a very strong shock), then the mesh cells will acquire locally a high aspect ratio, which our mesh-quality control motions will try to eliminate, if needed by breaking the mesh symmetry (cell shapes are made "round" through small transverse motions). This is what happens when we start the Couette flow impulsively on a non-suitable mesh. The introduction of asymmetries in the mesh can influence the flow, in particular in situations where fluid instabilities develop (see the Kelvin Helmholtz instability test in Springel, 2010a), where such asymmetric discretization errors can source growing perturbations. We note however that also on regular Cartesian meshes similar "grid-sourcing" errors exists. It appears unlikely that the poorer ability of the dynamic Voronoi mesh to maintain strict mesh symmetry is particularly detrimental for physical applications.

3.4.5 Flow Past a Circular Cylinder

We next consider the flow over a circular cylinder immersed in a wind tunnel. The geometric setup of the problem is shown in Fig. 3.17. The flow comes from the left at a fixed horizontal velocity U. The upper and lower boundaries are also kept at constant velocity U. Far from the cylinder, at the right end of the computational domain, we impose again an exit velocity U. The injection and exit regions are forced to have the prescribed inflow and outflow mass fluxes at all times, something that we numerically impose through small "buffer" regions as labeled in Fig. 3.17. For static Cartesian grids, this buffer region does not need to extend more than one cell in the x-direction. However, moving grids require not only the injection of momentum from the left, but also the injection of new mesh-generating points, since the wind tunnel will otherwise produce a depletion of cells at the left end as the mesh generating points drift to the right in the direction of the flow. We address this issue by letting the mesh automatically generate new cells through cell splitting, as already implemented in AREPO (see examples in Springel, 2010a). In doing this, some attention must however be paid to guarantee that the new cells reproduce the externally imposed inflow boundary conditions, which is most easily achieved with a sufficiently broad buffer region on the left end of the wind tunnel that covers the region where new cells are injected. Similarly, we employ the ability of AREPO to automatically remove mesh cells to prevent them from piling up on the right end of the wind tunnel. Altogether, we have created a wind tunnel that is filled with a mesh that blows with constant velocity from left to right, in a quasi-stationary state.

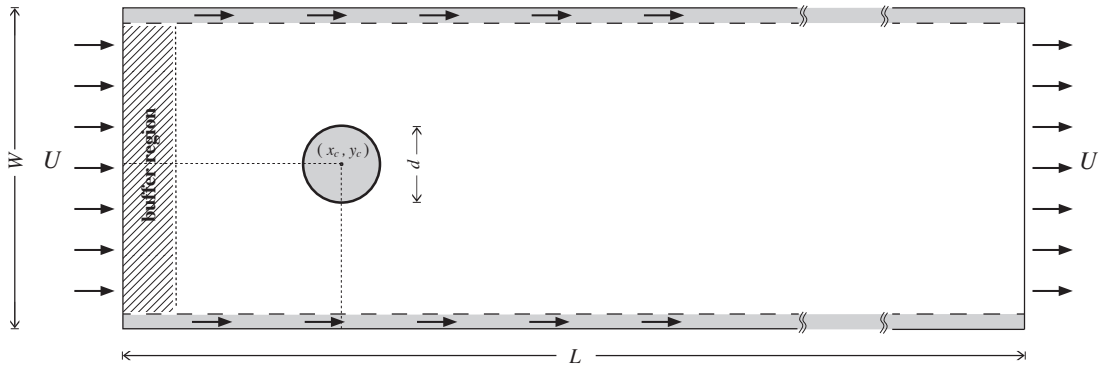


Figure 3.17.— Geometry of our wind tunnel set-up with a circular obstacle.

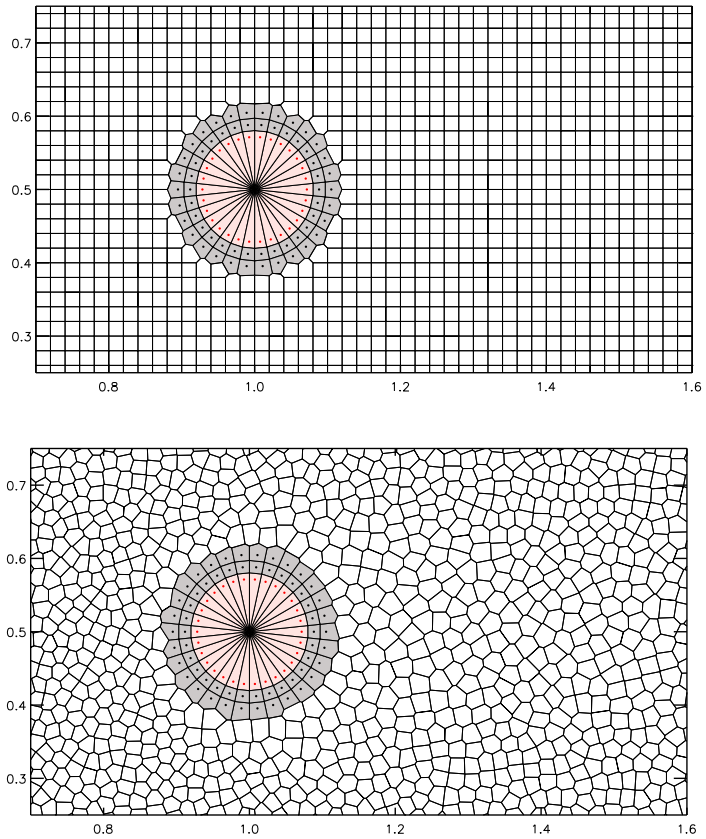


Figure 3.18.— Mesh near a circular cylinder inside a wind tunnel. The mesh contains both stationary mesh-generating points (defining the solid cylinder and two layers of cells used to create the cylindrical solid surface) and moving mesh-generating points (the remainder of the grid). The upper panel shows the initial setup, which highlights the cells representing the solid cylinder, and the two layers of fluid cells for which the equations of hydrodynamics are solved as in a standard stationary mesh. The total number of cells in the wind tunnel is $12,478$ (roughly 250×50). The perimeter of the cylinder is outlined by 30 cells, and its diameter is equivalent to eight cells across. The Voronoi faces in between the red and grey cells define the boundary at which the no-slip condition is imposed. The lower panel shows the same region at a later time. Whereas one layer of mesh-generating points surrounding the cylinder has remained stationary, the rest of the background mesh has moved downstream and transformed to a generic unstructured Voronoi mesh as it moves along with the fluid.

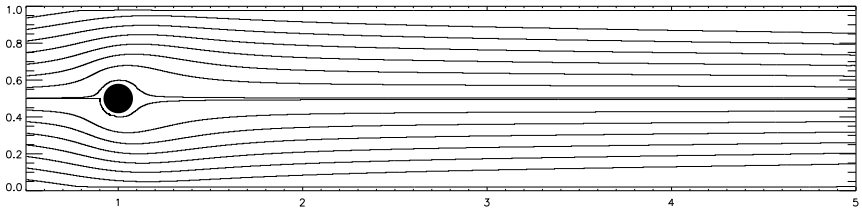
The other geometric parameters of the test problem we simulate here are the diameter d of the cylindrical obstacle, the width W of the tunnel and its length L. We have chosen $W = 6.25\,d$ and $L = 5\,W = 31.25d$, and have scaled all length units such that $W = 1.0$. The flexibility of the Voronoi mesh allows us to easily embed a cylindrical obstacle within the initially Cartesian background grid that fills the tunnel. Fig. 3.18 shows how we can tailor the mesh construction to reproduce the curved surface of the cylinder, using techniques similar to those that we used for the circular Couette flow problem.

The physical properties of the problem are primarily determined by the external velocity of the flow, U, and the dynamic viscosity of the fluid μ. In our numerical experiments we set the external flow velocity to $U = 0.5$, and combine this with constant initial pressure and density ($\rho = P = 1$). We take the fluid to be described by an ideal gas equation of state with adiabatic index $\gamma = 5/3$. The characteristic Reynolds number of the problem can then be defined by

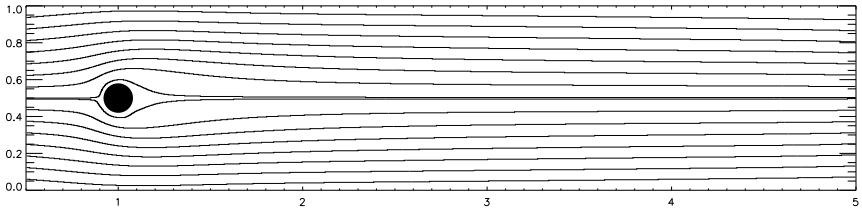
$$Re = \frac{U\,d}{\nu} = \frac{U\,d\,\rho}{\mu} \tag{3.65}$$

where ρ might however vary in time and space since the flow is fully compressible.

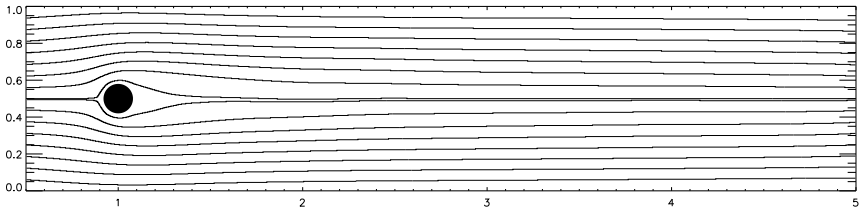
We have performed several numerical experiments of this problem using the viscous module added to AREPO. In each of these simulations, the Reynolds number is the only relevant quantity being changed. This is accomplished by changing μ exclusively, while keeping the other parameters fixed. Fig. 3.18 (upper panel) shows the initial setup for all the runs, which consist of a circular cylinder plus a Cartesian background grid of 250×50 mesh generating points. The dynamic viscosity coefficient μ takes five different values: 2.5×10^{-2}, 5×10^{-3}, 2.5×10^{-3}, 1.25×10^{-3} and 8.3×10^{-4}. These values



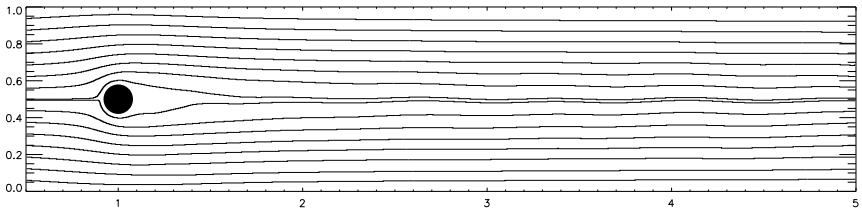
(a) $Re = 2$



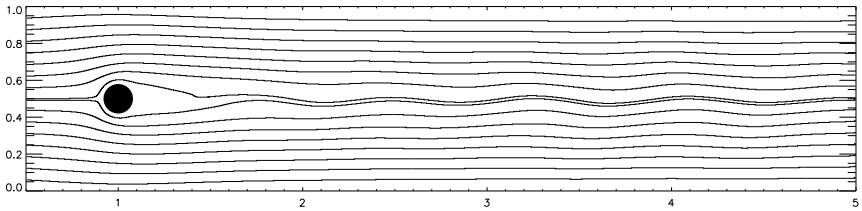
(b) $Re = 10$



(c) $Re = 20$



(d) $Re = 40$



(e) $Re = 60$

Figure 3.19.— Streamlines for compressible gas flow around a cylinder at five different Reynolds numbers, as labeled.

correspond to Reynolds numbers of 2, 10, 20, 40 and 60.

For each one of the tests, we show the resulting streamlines at time $t = 9.9$ (or an equivalent dimensionless time of $\bar{t} = tU/d \approx 31.0$) in Fig. 3.19. Below $Re \sim 40$, the flow is steady and symmetric above and below the cylinder. As the Reynolds number increases, the size of the wake behind the cylinder grows. Although in this example the structure of the wake is poorly resolved, the increase in Re is accompanied by an increase of vorticity confined within the wake.

Above $Re \sim 40$, the wake behind the cylinder starts to become unstable. This can be clearly seen in the streamline pattern of the $Re = 60$ panel. As the wake becomes unstable, the symmetry between the upper and lower portions of the domain is broken, at which point the flow becomes unsteady, such that the streamlines are no longer a valid representation of the Lagrangian trajectories of fluid parcels. This marks the onset of the *von Karman vortex street*, and the eventual transition to fully developed turbulence.

To further illustrate the flexibility of the mesh construction in AREPO, we can repeat this experiment with the mesh generating points set to remain static, thus recovering an Eulerian grid code. In addition, we increase the resolution by a factor of four to better resolve the wake behind the cylinder. In Fig. 3.20, we show the density contrast for five different Reynolds numbers. For stationary flow, the density distribution traces the streamline topology. At $Re = 100$, we can appreciate how the fully developed von Karman vortex street looks for a compressible gas.

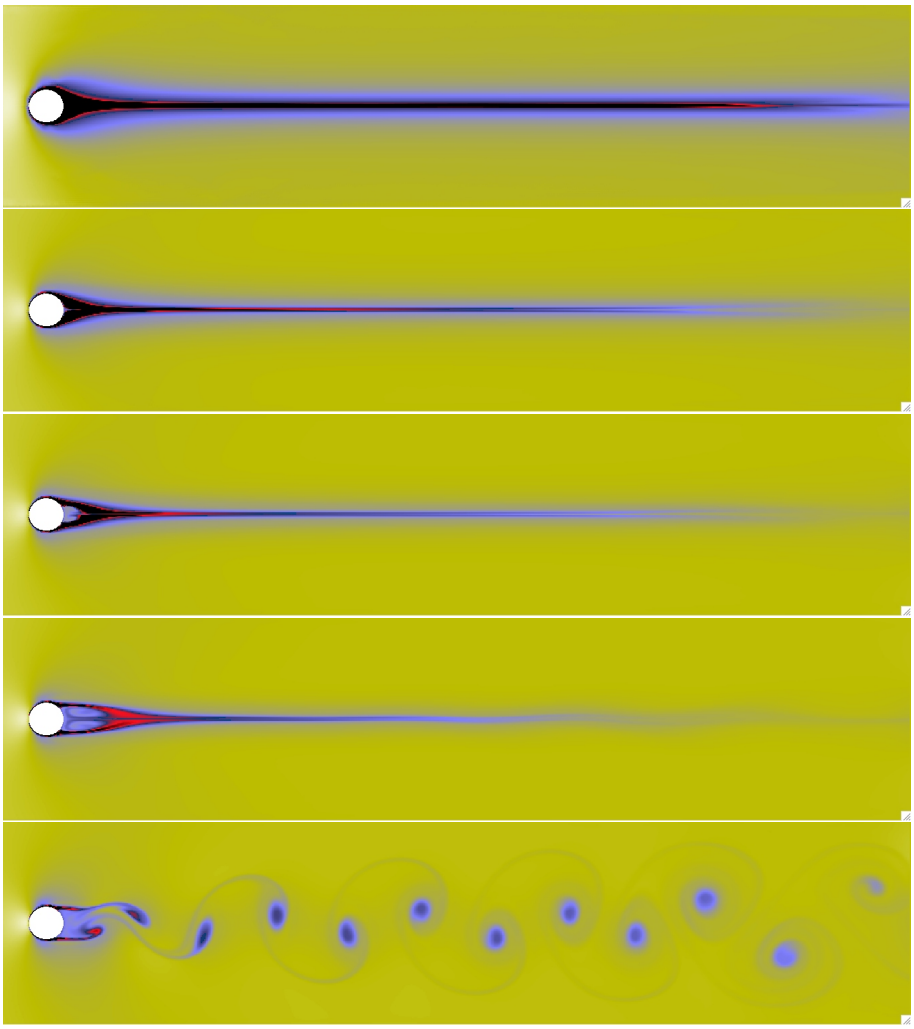


Figure 3.20.— Density contrast of compressible flow past a cylinder at five different Reynolds numbers, corresponding to $Re = 2$, 10, 20, 40, 100, from top to bottom. All five numerical experiments were computed with a static Cartesian mesh at moderately high resolution (1000×250), where the cell size is 1/32 of the cylinder's diameter.

3.4.6 Three Dimensions: Taylor-Couette Flow

Circular Couette flow is a stable, special case of the more complex and richer three-dimensional Taylor-Couette flow (Taylor, 1923). Taylor found that when the angular velocity of the inner cylinder is increased above a certain threshold, Couette flow becomes unstable. After this transition, different states have been identified, the most famous of which is the Taylor vortex flow, characterized by axisymmetric toroidal vortices. The diversity of states for Taylor-Couette flow has been explored in the past, most notably by Coles (1965) and Andereck et al. (1986). The latter work lists up to 18 different flow regimes observed in flow between independently rotating cylinders. Its "Andereck diagram", which explores the stability of the Taylor-Couette problem for a variety of Reynolds numbers, has become the standard benchmark for computational experiments of flow between rotating cylinders.

Although the computational and experimental study of three-dimensional Couette flow peaked during the 1980's with the classical works of Andereck et al. (1986) and Marcus (1984a,b), in recent years it has regained popularity (e.g. Dong, 2007; Avila et al., 2008; Meseguer et al., 2009a,b) mainly driven by the experimental studies of magnetized and unmagnetized rotating flows of Ji et al. (2001, 2006) and Sisan et al. (2004), which have resulted in significant progress on the characterization of the magnetorotaional instability (MRI; Balbus & Hawley, 1998) in the laboratory.

In this section, we briefly explore the evolution of Taylor-Couette flow on a moving Voronoi mesh. Although the AREPO code is not specifically designed for problems with symmetric geometries where static cylindrical meshes have proven to be more suitable, we have included this test to emphasize that our method works in three dimensions in

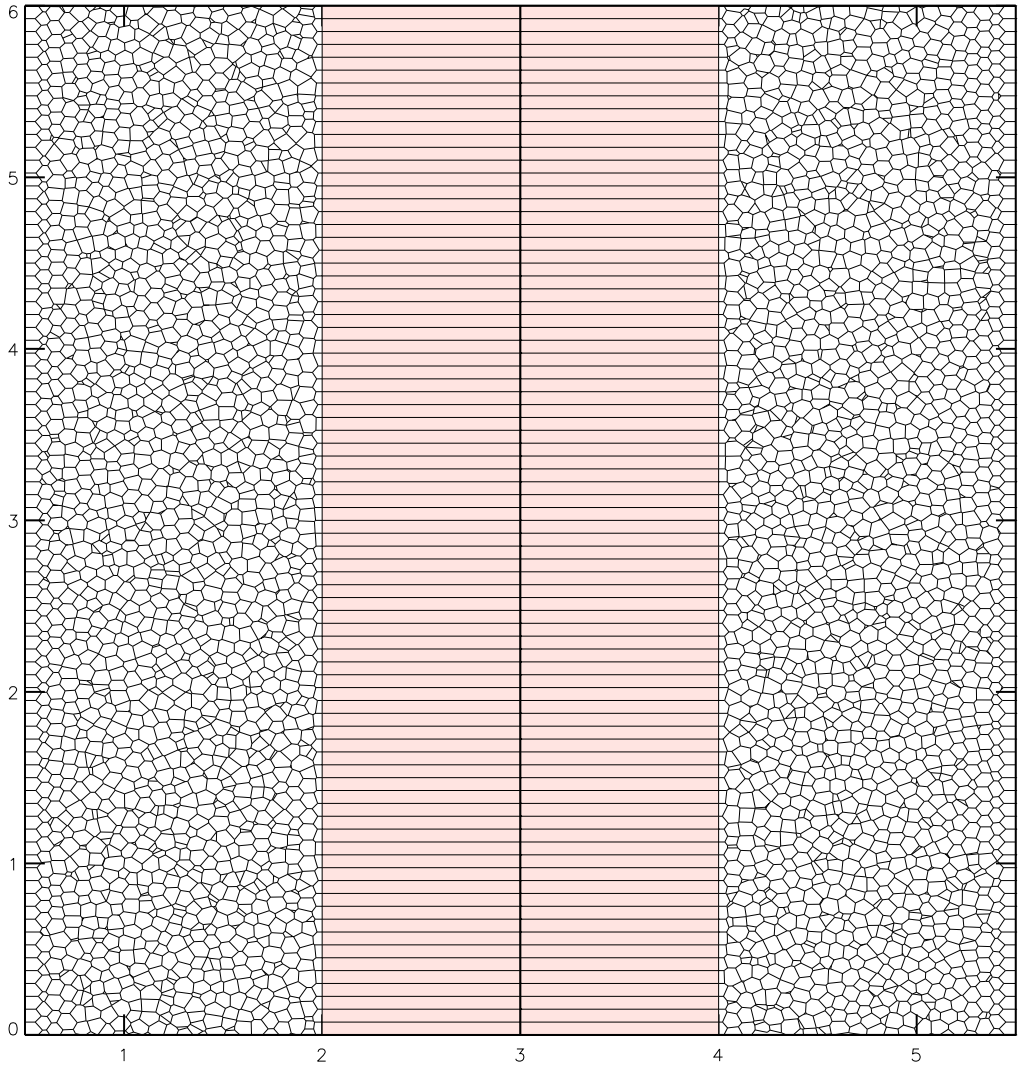
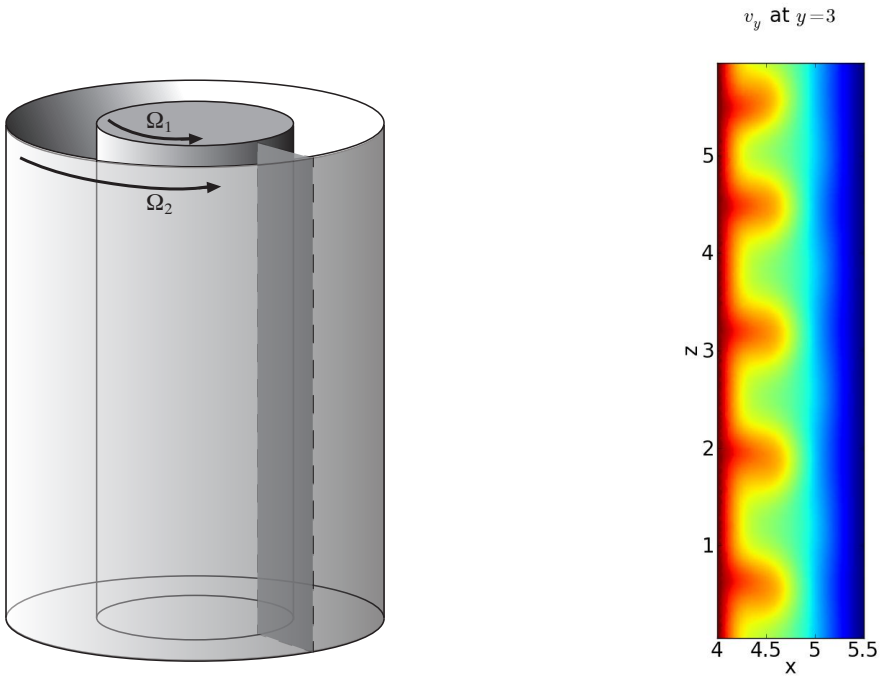


Figure 3.21.— Vertical slice of the three-dimensional Voronoi tessellation in Taylor-Couette flow at the time Taylor vortices have developed. This same slice is used when visualizing the v_x,v_y and v_z fields (Fig. 3.22b and Fig. 3.23).

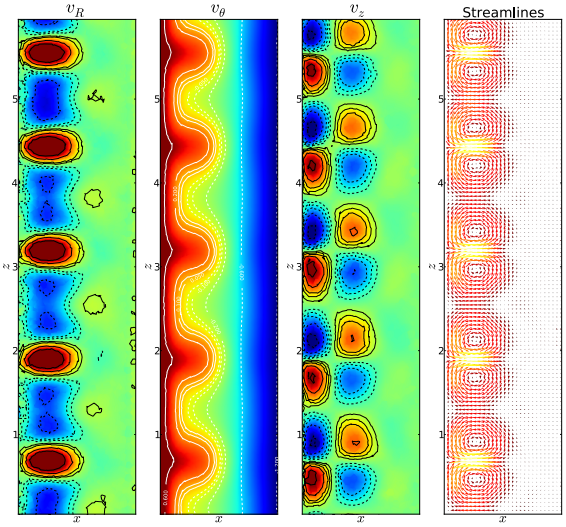
an analogous way to the two-dimensional examples shown above. It is straightforward to extend the two-dimensional Couette flow shown above to three dimensions using AREPO. Since the mesh is obtained from a distribution of mesh-generating points, all that is needed is to replicate the initial conditions shown in Figure 3.15 in the vertical direction (about 80 times) to fill up a cubic box.

A standard validation for a Taylor-Couette simulation with azimuthal and axial periodicity may include, for example (Avila, private communication): obtaining perfect axial symmetry at low Reynolds number (i.e. circular Couette flow), followed by obtaining the first bifurcation to axially symmetric Taylor vortices, and by reaching the second bifurcation to wavy vortices. These transitions occur sequentially as the angular velocity of the inner cylinder is increased while keeping the outer cylinder stationary (see the phase diagram of Andereck et al., 1986). However, it is not the purpose of this section to explore these transitions exhaustively; we only want to show that the third dimension works with our technique. We thus have focused on a particular configuration: counter-rotating Taylor-Couette flow, for which it is easy to obtain axially symmetric Taylor vortices (although these might relax back to Couette flow after several rotation periods; e.g. Liao et al., 1999). Since a faithful comparison to the benchmark results of Andereck et al. (1986), is out of the scope of this paper, we simply replicate the geometry described in Fig. 3.13 in the vertical direction such that the computational domain is now a cube of dimensions $6 \times 6 \times 6$, with periodic boundary conditions in the z-direction. The initially Cartesian mesh will eventually relax in all directions as the flow evolves (Fig. 3.21). The cylinder is effectively infinite, like in the two-dimensional case, except that this time there is no imposed symmetry along the z-direction. We choose the cylinder radii to be $R_1 = 1.0$ and $R_2 = 2.5$, just like in the 2D example, and

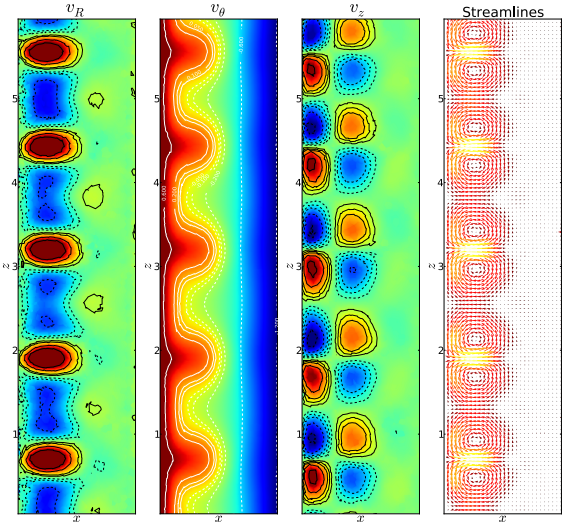


(a) (b)

Figure 3.22.— a) Illustration of the three-dimensional flow between two independently rotating cylinders. The figure shows a plane along the radial direction where the local velocity field is evaluated. b) Velocity field in the y-direction between the cylinders for a slice defined by $y = 3$ (i.e. along the diameter of both cylinders). For this particular plane, v_y is equivalent to the azimuthal velocity v_θ. The color scale goes from $v_y = \Omega_2 R_2 = -1.25$ (blue) to $v_y = \Omega_1 R_1 = 0.8$ (red). This example shows that v_θ is no longer independent of z. Thus the two-dimensional solution of Eq. (3.64) is no longer valid.



(a)



(b)

Figure 3.23.— Velocity structure of axisymmetric Taylor vortex flow at time $t = 119$ at two different meridian planes separated by 90°. The poloidal velocity field (v_R, v_z) is color mapped in the linear range [−0.08 (blue), +0.08 (red)], while the azimuthal field (v_θ) is color mapped in the linear range [−1.2 (blue), 0.6(red)]. The streamlines illustrate the vector field in the poloidal plane, showing with clarity the nature of Taylor vortices.

the respective angular velocities are $\Omega_1 = 0.8$ $\Omega_2 = -0.5$ (counterrotating). The dynamic viscosity is $\mu = 0.005$ and the fluid is started from rest with $\rho = P = 1$. The inner and outer Reynolds numbers ($Re_i \equiv R_i\Omega_i(R_2 - R_1)\rho/\mu$; e.g. Liao et al., 1999) are $R_1 = 240$ and $R_2 = -375$, respectively, where negative values indicate clockwise rotation.

The geometry of the problem is shown in Fig. 3.22a. A vertical slice is taken at a time when the Taylor vortices have developed (the corresponding sliced mesh is shown in Fig. 3.21). The azimuthal velocity on that slice shows deviations from the symmetry in z present in the circular Couette regime (Fig. 3.22b). Looking at the poloidal velocity field on that same slice (v_x and v_z in Fig. 3.23) one can appreciate, near the inner cylinder, the circular vertical motion characteristic of the Taylor vortices. This type of flow starts to develop at time $t \sim 60$ and remains essentially unaltered for several rotation periods. At much longer time scales, the flow could presumably decay back to a two-dimensional Couette flow as seen in the roughly similar test carried out by Liao et al. (1999), although we restate that the physical parameters used here are not necessarily equivalent to those in Liao et al. (1999) and Andereck et al. (1986).

In Figure 3.23, we show the velocity field of this Taylor-Couette experiment at time $t = 128$ for two different slices of the volume: (a) along the x-axis, and (b) along the y-axis (i.e. at 90° from the first slice). Except for the numerical noise, the two solutions are nearly indistinguishable, evidence of a global axially symmetric Taylor vortex flow (for a similar result, see Fig. 3 in Liao et al., 1999). Instead of taking velocity slices at particular values of ϕ, the axisymmetry can be tested further by directly averaging all velocity components along the azimuthal direction. The averaging procedure retains the flow morphology seen in Figure 3.23, without degrading it. We have measured the variability of v_R, v_ϕ and v_z along the ϕ-direction for different locations in the

R-z plane. For each of these quantities, we find that the rms fluctuations represent roughly $\sim 10\% - 15\%$ of the average value in regions of velocity maxima/minima. Most importantly, none of these fluctuations show systematic trends with the azimuthal coordinate.

3.5 Concluding Remarks

We have presented a new numerical approach for solving the three-dimensional, compressible NS equations on a dynamic mesh using the new astrophysical code AREPO. This novel approach, an extension of the finite volume method, defines the computational mesh as a Voronoi tessellation moving with the local flow. The advantages of using a dynamic Voronoi mesh for transient and stationary flows under diverse boundary conditions has been addressed. The implicit adaptivity of the quasi-Lagrangian mesh elements, in addition to the well-behaved topological properties of the Voronoi tessellation, ensure both geometric flexibility and low numerical diffusivity. In addition, the shock capturing, second-order-accurate finite-volume scheme implemented in the rest-frame of each moving cell provides high accuracy.

We have described in detail the algorithm used to estimate the viscous diffusion of momentum across inter-cell boundaries. Our scheme produces smoothly varying estimates of the viscous terms, resulting in accurate and stable solutions. The method extends previously known finite-volume formulations of the NS equations with the introduction of a new reconstruction scheme that represents a compromise between the use of piece-wise constant gradients and fully consistent quadratic-reconstruction schemes.

For pure hydrodynamic flow, the CPU time consumption of our code per timestep is typically quite a bit higher than for structured mesh codes or SPH codes, for the same number of resolution elements (see Vogelsberger et al., 2012, for a detailed comparison of CPU cost between AREPO and SPH). The additional computational time goes mostly into the Voronoi mesh construction overhead, which is simply not needed by a structured mesh code, and also into an enlarged computational cost for the flux computations. The latter comes about because of a larger number of faces per cell (in 3D, there are 6 sides for a cubical cell, but for a 3D Voronoi mesh, we have of order ~ 12 sides for each polyhedral cell). It is however important to note that other, problem-dependent factors should be taken into account when assessing the performance in practice. For example, if large bulk velocities are present, our method can take considerably larger timesteps than a corresponding fixed mesh code. Also, because the advection errors are reduced in our scheme, fewer cells are required to reach a given accuracy, so that our code can then end up being computationally more efficient. We also note that once self-gravity is included (as in many of our primary target applications in astrophysics), the relative speed difference in the hydrodynamic part between the structured fixed mesh and our moving Voronoi mesh becomes much less of an issue, because the cost of calculating self-gravity sufficiently accurately for arbitrary geometries substantially reduces the relative importance of the hydrodynamical cost.

As part of our study, we have verified the reliability of our new method through a series of example calculations that range from simple flows with known analytic solutions to traditional experiments of well-known quantitative behavior. The demonstrated ability of the scheme to reproduce exact solutions as a function of time, even if the flow is started impulsively from rest, is reassuring. These examples also show the flexibility

of the scheme in the presence of different solid surfaces moving in diverse ways. In all of these examples, the overall structure of the numerical scheme is identical, and the boundary conditions are set solely by the prescribed motion of the surfaces, which consist of controlled collections of Voronoi cells.

Although we have tested the performance of AREPO in configurations possessing a high degree of symmetry, it is in complex asymmetric problems where the moving-mesh approach would show all its power. The flexibility of the Lagrangian nature of the mesh will allow us to simulate, for example, complex astrophysical objects where viscosity is presumed to play a significant role. One such problem is the simulation of accretion disks around young stars. Although angular momentum transport in accretion disks is attributed to turbulence (most likely of magneto-hydrodynamic nature), this process is usually modeled both analytically (e.g. Shakura & Sunyaev, 1973; Lynden-Bell & Pringle, 1974; Pringle, 1981; Lin & Pringle, 1987) as well numerically (e.g. Kley & Lin, 1992; Masset, 2000; D'Angelo et al., 2002; de Val-Borro et al., 2006; Paardekooper & Mellema, 2006; Mudryk & Murray, 2009, just to name a few) by laminar flow in the presence of turbulent viscosity (Boussinesq approximation to eddy viscosity), due to the computational cost of global models of magneto-hydrodynamic disks. This kind of simulation would require further testing of the local and global conservation of angular momentum in AREPO, specially when compared to schemes that solve for the evolution of angular momentum directly (e.g. grid codes written in polar/cylindrical coordinates). Such tests will be the subject of future work.

Another application of viscous flow is the plasma viscosity at galaxy cluster scales (e.g. Sijacki & Springel, 2006). However, it is likely that in such systems viscosity, as well as thermal conduction, is anisotropic (Braginskii, 1965; see Dong & Stone, 2009

for an example). In such a case, the viscous stress tensor in Eq. (3.6) can be easily generalized to include the up to seven independent viscosity coefficients (Lifshitz & Pitaevskii, 1981). It will be particularly exciting to couple the local anisotropy directly to the magnetic field topology, with the latter calculated self-consistently using a recent magnetohydrodynamics implementation in AREPO (Pakmor et al., 2011).

Its powerful flexibility will make AREPO an interesting code both for astrophysical simulations of viscous flow, but potentially also in engineering applications where the ability to cope with curved and moving boundaries is particularly attractive.

3.6 Appendix: Gradient Extrapolation Coefficients

The extrapolation of the velocity gradients (e.g. Eq. 3.36) requires a numerical estimate of the gradient matrix as well as an estimate for the time derivative of the gradient. For the latter, the tensors $A_{\alpha\beta b}$ and $B_{\alpha\beta ba}$ are needed (Eq. 3.34). Both tensors depend on the cell-centered scalar quantities as well as their gradients. The values of $A_{\alpha\beta b}$ are (e.g. Toro, 2009)

$$A_{\alpha\beta x} = A_{\alpha\beta 1} = \begin{pmatrix} v_x & \rho & 0 & 0 & 0 \\ 0 & v_x & 0 & 0 & 1/\rho \\ 0 & 0 & v_x & 0 & 0 \\ 0 & 0 & 0 & v_x & 0 \\ 0 & \gamma P & 0 & 0 & v_x \end{pmatrix}, \tag{3.66}$$

$$A_{\alpha\beta y} = A_{\alpha\beta 2} = \begin{pmatrix} v_y & 0 & \rho & 0 & 0 \\ 0 & v_y & 0 & 0 & 0 \\ 0 & 0 & v_y & 0 & 1/\rho \\ 0 & 0 & 0 & v_y & 0 \\ 0 & 0 & \gamma P & 0 & v_y \end{pmatrix}, \tag{3.67}$$

$$A_{\alpha\beta z} = A_{\alpha\beta 3} = \begin{pmatrix} v_z & 0 & 0 & \rho & 0 \\ 0 & v_z & 0 & 0 & 0 \\ 0 & 0 & v_z & 0 & 0 \\ 0 & 0 & 0 & v_z & 1/\rho \\ 0 & 0 & 0 & \gamma P & v_z \end{pmatrix}. \tag{3.68}$$

The tensor $B_{\alpha\beta ba} \equiv \partial_a A_{\alpha\beta b} = A_{\alpha\beta b,a}$ (with $a, b = x, y, z$ or $1, 2, 3$ and $\alpha, \beta = 0, 1, 2, 3, 4$) has components:

$$B_{\alpha 0 xa} = \begin{pmatrix} \partial_x v_x & \partial_y v_x & \partial_z v_x \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{3.69}$$

$$B_{\alpha 0 ya} = \begin{pmatrix} \partial_x v_y & \partial_y v_y & \partial_z v_y \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{3.70}$$

$$B_{\alpha 0za} = \begin{pmatrix} \partial_x v_z & \partial_y v_z & \partial_z v_z \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} , \tag{3.71}$$

$$B_{\alpha 1xa} = \begin{pmatrix} \partial_x \rho & \partial_y \rho & \partial_z \rho \\ \partial_x v_x & \partial_y v_x & \partial_z v_x \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \gamma \partial_x P & \gamma \partial_y P & \gamma \partial_z P \end{pmatrix} , \tag{3.72}$$

$$B_{\alpha 1ya} = \begin{pmatrix} 0 & 0 & 0 \\ \partial_x v_y & \partial_y v_y & \partial_z v_y \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} , \tag{3.73}$$

$$B_{\alpha 1za} = \begin{pmatrix} 0 & 0 & 0 \\ \partial_x v_z & \partial_y v_z & \partial_z v_z \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} , \tag{3.74}$$

$$B_{\alpha 2xa} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \partial_x v_x & \partial_y v_x & \partial_z v_x \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} , \tag{3.75}$$

$$B_{\alpha 2ya} = \begin{pmatrix} \partial_x \rho & \partial_y \rho & \partial_z \rho \\ 0 & 0 & 0 \\ \partial_x v_y & \partial_y v_y & \partial_z v_y \\ 0 & 0 & 0 \\ \gamma \partial_x P & \gamma \partial_y P & \gamma \partial_z P \end{pmatrix} , \tag{3.76}$$

$$B_{\alpha 2za} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \partial_x v_z & \partial_y v_z & \partial_z v_z \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} , \tag{3.77}$$

$$B_{\alpha 3xa} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \partial_x v_x & \partial_y v_x & \partial_z v_x \\ 0 & 0 & 0 \end{pmatrix} , \tag{3.78}$$

$$B_{\alpha 3ya} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \partial_x v_y & \partial_y v_y & \partial_z v_y \\ 0 & 0 & 0 \end{pmatrix} , \tag{3.79}$$

$$B_{\alpha 3za} = \begin{pmatrix} \partial_x \rho & \partial_y \rho & \partial_z \rho \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \partial_x v_z & \partial_y v_z & \partial_z v_z \\ \gamma \partial_x P & \gamma \partial_y P & \gamma \partial_z P \end{pmatrix} , \tag{3.80}$$

$$B_{\alpha 4xa} = \begin{pmatrix} 0 & 0 & 0 \\ -\dfrac{\partial_x \rho}{\rho^2} & -\dfrac{\partial_y \rho}{\rho^2} & -\dfrac{\partial_z \rho}{\rho^2} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \partial_x v_x & \partial_y v_x & \partial_z v_x \end{pmatrix} , \tag{3.81}$$

$$B_{\alpha 4ya} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -\dfrac{\partial_x \rho}{\rho^2} & -\dfrac{\partial_y \rho}{\rho^2} & -\dfrac{\partial_z \rho}{\rho^2} \\ 0 & 0 & 0 \\ \partial_x v_y & \partial_y v_y & \partial_z v_y \end{pmatrix} , \tag{3.82}$$

$$B_{\alpha 4za} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -\dfrac{\partial_x \rho}{\rho^2} & -\dfrac{\partial_y \rho}{\rho^2} & -\dfrac{\partial_z \rho}{\rho^2} \\ \partial_x v_z & \partial_y v_z & \partial_z v_z \end{pmatrix} . \tag{3.83}$$

Part II: Circumstellar Disk Simulations on a Moving Voronoi Mesh

4

Planet-disk Interaction on a *Freely* Moving Mesh

4.1 Introduction

Mesh-construction is a fundamental step in the process of solving sets of partial differential equations numerically. Although a mesh itself is an "extraneous" element to the underlying equations of hydrodynamics (i.e., there is nothing in the theory that can tell us which is the "correct" way to discretize space), a given choice of mesh can have a significant impact on the results of a numerical experiment or numerical simulation. This is of particular importance in astrophysics, where gas-dynamical flows are often *far* from being entirely resolved and therefore sensitive to the physics at the resolution scale. In practice, the choice of the mesh can have as much of an impact in the numerical result as other elements of the computational methodology, such as the adequacy of the coordinate frame used, the order-of-accuracy of the scheme, or even the additional sub-resolution

models implemented. Given the degeneracy between the mesh and other "features" of a numerical code, the choice of the optimal discretization approach (whether it is through a grid-approach or a particle-based approach) tends to depend on the problem at hand.

Recently, moving-mesh methods for computational gas dynamics (Springel, 2010a; Duffell & MacFadyen, 2011; but see also Borgers & Peskin, 1987; Trease, 1988; Dukowicz et al., 1989), as well as novel mesh-less approaches like that of McNally et al. (2012b), have been shown to be an interesting and powerful tool for studying high-mach-number, large-dynamical-range astrophysical flows.

High-mach-number flows are computationally challenging for several reasons. The most common complication is the so-called "high-Mach number problem" (Ryu et al., 1993; Bryan et al., 1995; Feng et al., 2004; Trac & Pen, 2004). This problem manifests itself when the kinetic energy density is much larger than the internal/thermal energy density. Consequently a small fractional error in the velocity can translate into large fractional error in temperature, eventually distorting the thermodynamic evolution of the gas. Another problem is the strict limitation that the Courant time-step condition imposes on high-velocity flows, which extends the computation time beyond practicality, with the additional peril of excessive numerical noise accumulated over a large number of integration time-steps.

One such system in which high-Mach number flows pose a serious computational limitation is that of astrophysical accretion disks. In these systems, extremely small Courant-limited time-steps are one of the greatest barriers to high-resolution simulations. In addition, the physics of these systems usually evolves over many dynamical/orbital times (hundreds, thousands or more), making simulation studies extremely expensive.

The azimuthal Mach number of a disk is roughly the Keplerian speed divided by the local sound speed $\mathcal{M}_\phi \sim v_K/c_s \sim h$, where the aspect ratio h of a thin disk is $0.01 - 0.1$. Thus, Mach numbers can reach values of several tens. The short time-steps implied by these high-velocity azimuthal flows is one of the reasons that during the first decade of modern computer simulations of accretions disks, most numerical experiments were run in the "local" or "shearing sheet/box" (e.g., Hawley & Balbus, 1991; Hawley et al., 1995; Stone et al., 1996) approximation of Goldreich & Lynden-Bell (1965) (see also Narayan et al., 1987).

The importance and ubiquitousness of accretion disks in astrophysics has propelled the development of several (magneto-) hydrodynamics codes specifically written for the numerical evolution of global models of disks. Global disk simulations over hundreds of orbits became feasible when the FARGO (Fast Advection in Rotating Gaseous Objects Masset, 2000) algorithm was introduced. In this scheme, the logic of the shearing-sheet approximations was applied to global simulations: i.e., the Euler equations are solved in a non-inertial rotating frame. To first order, the disk is rotating at Keplerian speed, thus the equations can be written in the local non-inertial frame and the Courant criterion is thus based off the deviation from the background velocity in this non-inertial frame, and not off the highly supersonic speed as measured from the inertial frame. Besides the FARGO code, schemes like RH2D (Kley, 1989) RODEO (Paardekooper & Mellema, 2006), RAPID (Mudryk & Murray, 2009) and DISCO (Duffell & MacFadyen, 2013), among others, have been tailored specifically for the solution of the Euler equations in cylindrical/polar coordinates for supersonic Keplerian flow.

Some of these cylindrical-coordinate codes (e.g., RODEO) were specifically designed to target the problem of planet-disk or satellite-disk interaction, namely, the tidal

interaction between a disk and an embedded planet in Keplerian rotation. This gravitational coupling between a planet and a gas disk has been studied in detail theoretically Goldreich & Tremaine, 1979, 1980; Lin & Papaloizou, 1979, 1986a,b; Ward, 1986; Takeuchi et al., 1996; Ward, 1997; Tanaka et al., 2002 and more recently Rafikov & Petrovich, 2012 and Petrovich & Rafikov, 2012) as well as computationally (e.g., Bate et al., 2003; de Val-Borro et al., 2006; D'Angelo & Lubow, 2008, 2010; Dong et al., 2011b,a; Duffell & MacFadyen, 2013).

Planet-disk interaction bears direct relevance to the early dynamical evolution of protoplanets and proto-planetary systems, since it is the mechanism behind disk-induced planet migration (e.g. Ward, 1986, 1997). The fully nonlinear integration of this problem through numerical simulation is essential for understanding the gravitational coupling between the planet and the surrounding disk, especially if the planet is to open a gap. Gaps are not only important for the dynamics of planet migration, but they may also provide morphological clues about the presence of young planets around stars in observations of circumstellar disk. The existence of central cavities (Calvet et al., 2002; Andrews et al., 2011) and even gaps (e.g., Espaillat et al., 2007) in gas disks, inferred from both imaging and infrared spectroscopy, opens the tantalizing possibility that the disk structure is being disturbed by an embedded planet. Observational efforts are ongoing to directly detect the planets responsible for gaps and cavities (see, for example Kraus & Ireland, 2012).

Although the process of gap-opening by a single planet in a circular orbit embedded in a two-dimensional, isothermal disk is far too idealized to be a realistic model for planet-disk interaction, it remains an important basic test for hydrodynamic codes. The complexity of the non-linear interaction between a planet and a disk does not allow for

traditional code *verification* (i.e. there is no known *exact* solution), but the number of numerical experiments available in the literature allow for the very important task of code *benchmarking*. Most notably, de Val-Borro et al. (2006) carried out an extensive comparison of different grid-based and particle-based codes, identifying similarities and disagreements between different numerical schemes of widespread use in computational astrophysics. Along the lines of that code comparison project, we apply the moving-mesh scheme to the problem of planet-disk interaction in the gap-opening regime.

In this work we use the code AREPO (Springel, 2010a) for the first time in numerical experiments of planet-disk interaction. This problem has been studied in the low-planet mass case by Duffell & MacFadyen (2012) using the *structured* moving-mesh code DISCO. To our knowledge, an entirely *freely* moving mesh based on a Voronoi tessellation has not been applied successfully before to the type of problem at hand.

Moving-mesh methods share some of the spirit of the FARGO scheme, which is to solve the Euler equations in a frame moving with the local flow in order to bypass the restrictions imposed by the Courant criterion. Of course, one important difference is that a code like AREPO does not assume a priori the geometry and magnitude of the underlying density and velocity fields, and thus it becomes a quantity that varies from cell to cell that needs to be updated at every time-step. The only robust way to enable cells to move self-consistently and *freely* with the flow is to allow the distribution of cells to determine the mesh-topology through an automated tessellation. Although the Voronoi tessellation is not the only way to tessellate space (the Delaunay tessellation is a popular approach in computational fluid dynamics) it is –to our knowledge– the only one that changes continuously with the displacement of the generating points. This allows for a quasi-Lagrangian interpretation of the tessellation: at high resolutions, each

cell is a parcel of gas that follows the Lagrangian trajectories of the flow with a minimal change of its gas content.

Code comparisons between different grid- and particle-based methods abound in the astrophysics literature. Besides the aforementioned work of de Val-Borro et al. (2006), many of these studies have been oriented toward the methods of computational cosmology (e.g., Frenk et al., 1999; O'Shea et al., 2005; Heitmann et al., 2005; Agertz et al., 2007). Recently, Vogelsberger et al. (2012), Sijacki et al. (2012) and Kereš et al. (2012) carried out a comparison between smoothed particle hydrodynamics (SPH; using the GADGET code, Springel, 2005) and the moving-mesh code AREPO for simplified cosmological simulations as well as for idealized, isolated setups, isolating some of the differences and advantages moving-mesh codes have over the particle-based approach of SPH.

Recently, some researchers have raised concerns and caveats about the negative effects of having a freely moving-mesh. Although there is important merit to these concerns, we believe some of the limitations observed are exclusive to specific hydrodynamical schemes of other tessellation codes and not inherent to moving-mesh codes in general. In this work, we find that AREPO is qualitatively competitive with polar grid codes in simulations of planet-disk interaction, and that there is no reason to consider the moving-mesh approach as fundamentally ill-suited to capture differentially rotating flows accurately.

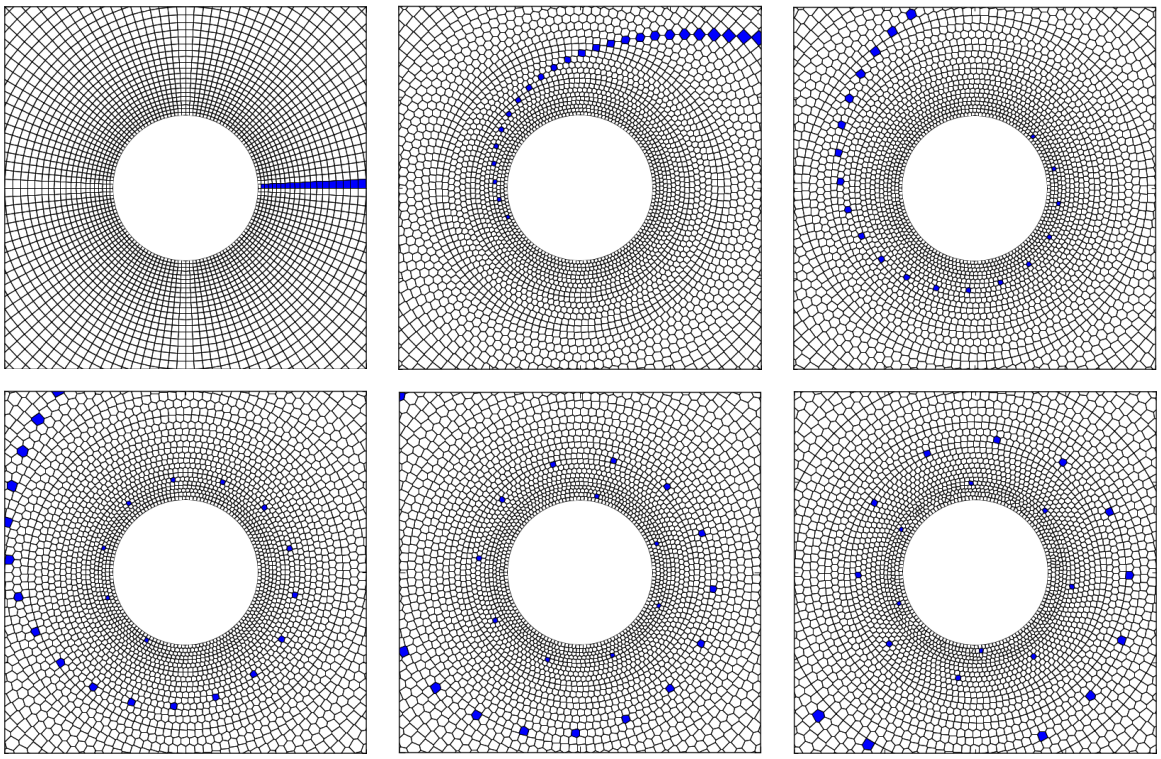


Figure 4.1.— Evolution of a Voronoi mesh under differential rotation supported by a Keplerian potential (time increases left-to-right and top-to-bottom). The mesh-generating points are initially positioned in a polar distribution (logarithmic spacing in radius), which is roughly maintained. The color-filled cells correspond to set of cells tagged according to ID number at time= 0(top left panel) and subsequently followed in time. The spatial distribution of the tagged cells highlights the "quasi-Lagrangian" nature (Vogelsberger et al., 2012) of the moving-mesh approach in the case of Keplerian shear.

4.2 Numerical Experiments

4.2.1 Problem Setup

Hydrodynamic Equations. In this work we focus on the solution of the Euler equations in two dimensions. In conservation-law form, these equations are:

$$\frac{\partial}{\partial t}\Sigma + \nabla \cdot (\Sigma \mathbf{v}) = 0 \tag{4.1a}$$

$$\frac{\partial}{\partial t}(\Sigma \mathbf{v}) + \nabla \cdot (\Sigma \mathbf{v} \otimes \mathbf{v} + P\mathbf{I}) = -\Sigma \frac{\partial \Phi}{\partial \mathbf{r}} \tag{4.1b}$$

Note that the gravitational potential on the right hand side of Equation 4.1b is included as a source term. This is one of the differences to some of the grid-based codes written in cylindrical coordinates. In those coordinates, the Keplerian term can be included directly into the conservation laws since the radial gradient can be written as a part of the divergence term in the hyperbolic equations (see Kley, 1998; Paardekooper & Mellema, 2006).

The fact that the gravity force is not included in a manifestly conservative formulation into the Euler equations (however, see Springel, 2010a and Jiang et al., 2013 for alternative approaches to enforce a "flux-based" description of gravity) implies that a gravitational time-step criterion must be considered in addition to the Courant-criterion time step. This acceleration must be taken into account for cells to follow accurate orbital trajectories (Figure 4.1). The bulk orbital motion of the cells is carried out by a conventional kick-drift-kick (KDK) operator (e.g. Saha & Tremaine, 1992; Preto & Tremaine, 1999) which "brackets" the hydrodynamical finite-volume operation (Springel, 2010a). Although the concept of symplectic integrators such as the KDK carries little meaning when applied to non-reversible systems like compressible gas dynamics, it is

nonetheless true that, in practice, time-symmetric integrators have superior performance (in terms of energy and angular momentum conservation) than non-symmetric ones even when integrating orbits of SPH particles (Springel, 2005). Note that Duffell & MacFadyen (2011) use a Runge-Kutta (RK4) integration for the motion of the mesh-generating points. Although the RK4 is more accurate than the KDK leapfrog, it is well known to suffer from severe secular effects, while the leapfrog does not. If the disk is going to be evolved for hundreds and thousands of orbits, the choice of the mesh-drifting algorithm can be important.

Typically, the gravitational time-step will be shorter than the fluid frame Courant time step for low resolution runs. For high resolution, the Courant-time step is expected to dominate. This is because the orbital time-step should depend more weakly on cell size than the signal-crossing time. For KDK integrators of particles in Keplerian potential, about 50-100 time-steps should suffice to capture the orbit accurately. The acceleration time-step is always based on the local gravitational potential, such that it adapts to the planet potential when close to it, and is modified if self-gravity is included. As a result, AREPO could be more computationally expensive than the classic FARGO scheme, because time-steps are allowed to become shorter than the fluid-frame Courant time-step, and because the motion of the mesh needs to be *solved for* instead of being prescribed. In addition, the mesh needs to be re-tessellated at every time-step. Note that the concerns raised by Dong et al. (2011a) about the use of the fluid-frame Courant time-step ignoring the gravitational influence of the planet in FARGO should not be an issue in our case.

Gravitational Potential. Following de Val-Borro et al. (2006), we represent the star-planet system by an external, time varying potential:

$$\Phi(\mathbf{r},t) = -\frac{GM_*}{|\mathbf{r}|} - \frac{GM_p}{|\mathbf{r}-\mathbf{r}_p(t)|} + \frac{GM_p}{|\mathbf{r}_p(t)|^3}\mathbf{r}\cdot\mathbf{r}_p(t) \tag{4.2}$$

where the third term on the right hand side corresponds to the indirect term that results from choosing the coordinate system to be fixed to the central star. The planet's position vector is

$$\mathbf{r}_p(t) = a_p\cos\left(2\pi\, t/P_p\right)\hat{\mathbf{x}} + a_p\sin\left(2\pi\, t/P_p\right)\hat{\mathbf{y}} \quad , \tag{4.3}$$

(with $a_p = P_p = 1$) i.e. the planet moves in a circular orbit around the star.

The direct term corresponding to the planet potential (second term on the RHS of Equation 4.2) must be softened. We have chosen a spline-type gravitational softening for the planet potential as is usually done in GADGET (Springel et al., 2001). The spline softening ensures a smooth transition into the exact Newtonian potential at a finite distance from the planet (2.8 times the gravitational softening parameter). In here, we use a gravitational softening of $\epsilon = 0.03$ (0.6 times the disk scaleheight at the planet's position) in agreement with the general setup proposed by de Val-Borro et al. (2006) (see Dong et al., 2011a, for a discussion on the different types of softening and their effects).

In the presence of the planetary potential, the evolution of the mesh deviates from the nearly axially symmetric geometry of Figure 4.1 to one that adapts to the density evolution of the disk under the tidal torquing of the planet. Figure 4.2 shows the geometry of the mesh in our lowest resolution runs (see Section 4.2.2 below) for two different mass ratios after 100 planetary orbits. In the small mass ratio run, the distortion of the mesh is subtle but the characteristic spiral wake of the planet can be identified in the mesh as an overdensity of Voronoi cells. The runs with larger mass

ratios show a greater distruption to the original mesh; not only can one identify the spiral wake, but also one can observe a cell concentration near the planet and a cell under-density where the planet has cleared a gap. Outer concentrations produced by vortex instabilities are also visible.

Boundary Conditions. One advantage of the moving-mesh approach is the extreme flexibility for including moving boundaries *within* the computational domain (Springel, 2010a; Muñoz et al., 2013). The inner and outer boundaries of a circumstellar disc can be constructed from collections of mesh-generating points describing concentric cylinders/circles. These mesh-generating points move collectively, keeping the overall shape of the boundary as the tessellation is updated. This type of concentric boundary was already implemented in two- and three-dimensional simulations of a Couette flow between concentric cylinders by Muñoz et al. (2013, Chapter 3 in this thesis).

At the inner ($R_{\rm in} = 0.25$) and outer ($R_{\rm out} = 2.5$) boundaries we impose reflective boundary conditions (see Figure 4.3 for a detailed description). In this case, the primitive variables in "outside" cell (effectively treated as a ghost cell, although it exists within the computational domain) are copied from the adjacent "inside" cell, except for the velocity normal to the surface, which is reverted. The velocity gradients are kept the same, thus the velocity normal to the surface goes continuously through zero at the face, as required by reflective boundaries. The initial-value Riemann problem at this interface is solved in the same way as in the rest of the domain. To minimize wave reflections off the boundaries, we impose an absorbing layer or "wave-killing region de Val-Borro et al. (2006) that extends from the inner radius up to $R = 0.5$ and from the outer radius down to $R = 2.1$ thus reducing the self-consistent computational domain to the region between

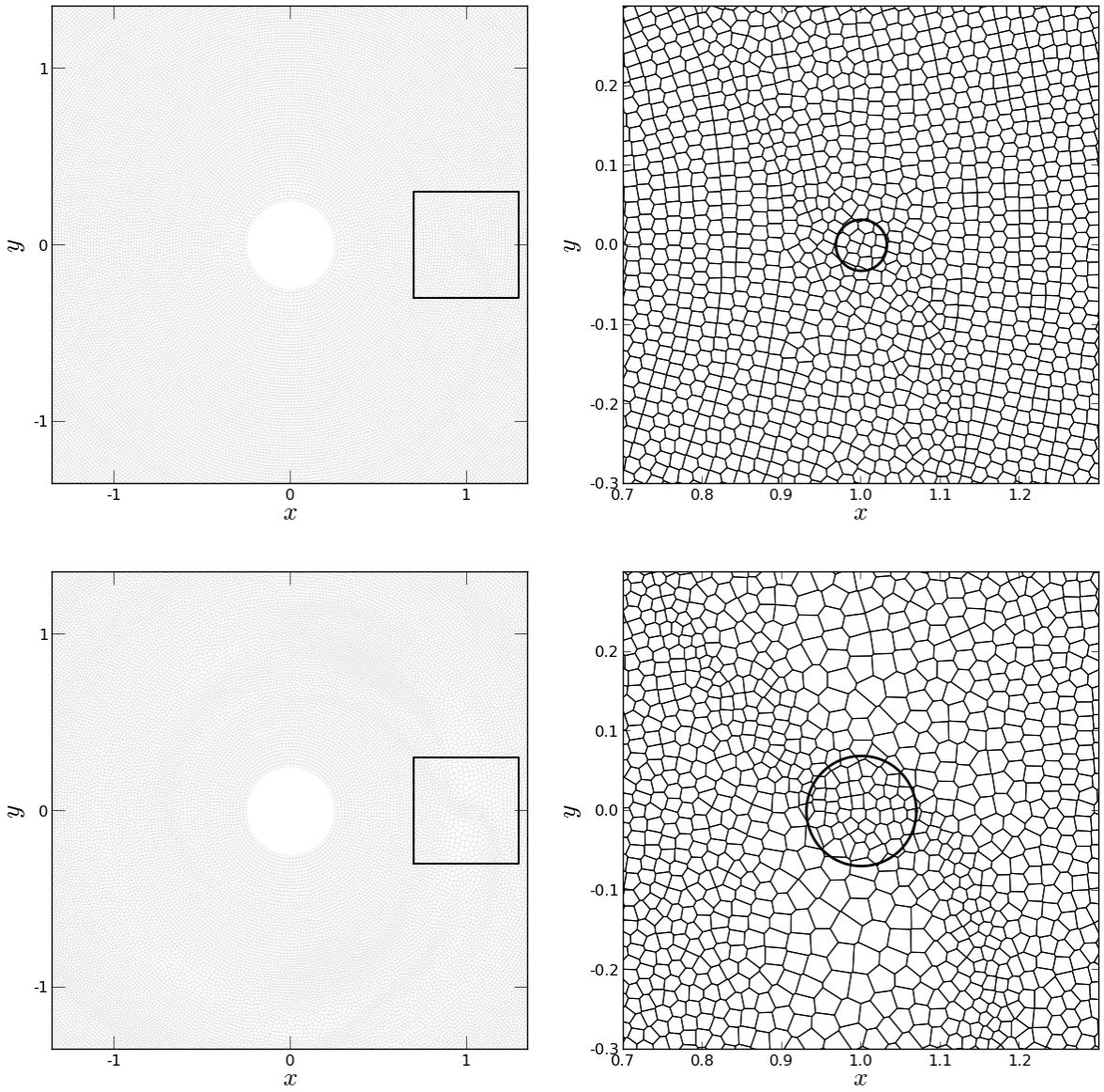


Figure 4.2.— Mesh geometry for two low resolution runs with mass ratios of $\mu = 10^{-4}$ (top panels) and $\mu = 10^{-3}$ (bottom panels). The insert in both rows shows a close-up to the mesh in the region close to the planet. Black circles represent the Hill sphere in both cases.

those radii. The absorbing region is implemented by adding a relaxation-like source term to the equations of motion, i.e.,

$$\frac{dX}{dt} = -\frac{X - X_0}{\tau}\Theta(R) \tag{4.4}$$

where X represents each primitive or conservation variable, X_0 is the reference value (the initial condition) and $\Theta(R)$ is a parabolic "ramp function" that vanishes at the edge of the absorbing region.

This approach is similar in spirit to the perfectly matched layer (PML) of Berenger (1994). In PMLs, the wave damping is obtained by modifying the equations of motions with frequency-dependent terms. The evanescence of the waves is enforced by introducing an artificial complex quantity into the dispersion relation of propagating waves, thus causing exponential decay of their amplitude when needed.

Shear Viscosity. A novel approach to physical viscosity has been recently developed by Muñoz et al. (2013, Chapter 3 in this thesis). This approach uses a new "hybrid double-linear" reconstruction scheme to cope with the truncation errors of the complex Voronoi cells while aiming to preserve the second order of the scheme in time and space. This approach uses estimates for high-order derivates to capture the spatial variability of the shear to overcome the difficulties that arise with a mesh where cells have irregular shapes and arbitrary number of neighbours. With this approach, the time-centred diffusion fluxes can be estimated at each interface.

Equation of state Circumstellar discs are often assumed to have an irradiation-dominated temperature structure (e.g. Chiang & Goldreich, 1997), and thus a non-evolving temperature specified by $T = T_0(R/R_0)^{-l}$. This is referred to as the "locally

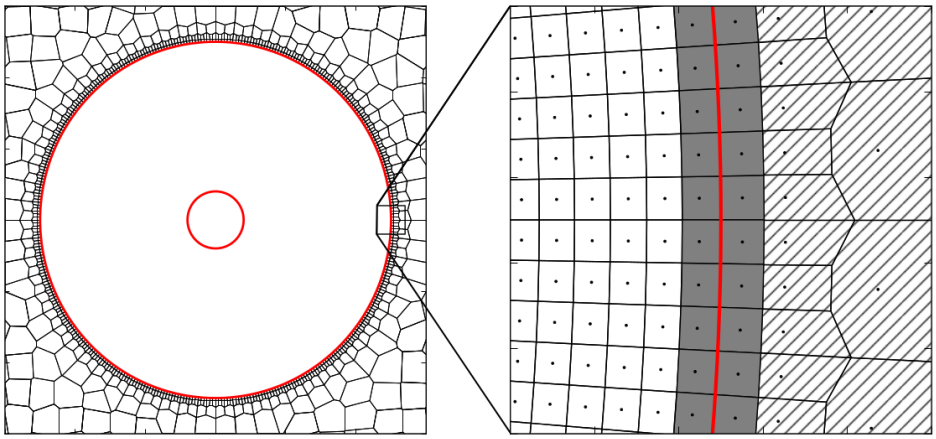


Figure 4.3.— Inner and outer boundaries (in red) for a two-dimensional circumstellar disk simulation. The computationally active domain – contained within the two boundaries – only represents a fraction of the disk, which is assumed to extend within the inner boundary and beyond the outer boundary. A background Voronoi mesh is added beyond the domain. The main purpose of these additional cells is to fill in the computational box. However, in this example, the background cells are "dead" and are never updated. A zoomed in portion of the mesh shows how each boundary is constructed using pairs of mesh-generating points placed following two concentric circumferences. Locally, the boundary is just one Voronoi interface and is, by definition, equidistant to both mesh-generating points. This boundary separates the interior domain from the exterior domain. The last interior cell and the first exterior cell are shown in gray. These cells are referred to as "inside" cell and "outside" cell, respectively (see text). Boundary conditions are imposed on the interface both at the moment of gradient estimation as well as after quantity extrapolation (Springel, 2010a). The background cells (cross-hatched region), although never updated, can be used as boundary values "at infinity" and used to solve for the hydrodynamics quantities of the outside cells if these values cannot be fully determined by the interior domain (see description of matching layer boundaries).

isothermal" approximation, and $l = 1$ is a common choice (see Kratter & Murray-Clay 2011 for a discussion of the validity of this assumption). The disk aspect ratio varies with radius as $h = h_p(R/R_p)^{(1-l)/2}$, where R_p is the radius of the planet's orbit and h_p is the disk aspect ratio at that location. Thus, disks with $l = 1$ have constant aspect ratio and disks with shallowed temperature profiles flare up with radius.

In shock-capturing Godunov schemes, this type of temperature structure requires the use of an isothermal Riemann solver, (e.g. Balsara, 1994), although a common shortcut is to run an adiabatic simulation with $\gamma = 1.001$. In this work, we make use of an iterative isothermal Riemann solver. For simplicity, we set the disk to have the same temperature *globally* (i.e. $l = 0$). This approximation departs from the fiducial model of de Val-Borro et al. (2006), which consisted of a locally isothermal disk of constant aspect ratio.

4.2.2 Initial Conditions

All the simulations presented in this paper are carried out in units where $G(M_* + M_p) = a_p = 1$ and thus $P_p = 2\pi$, where M_* is the mass of the central object, M_p, the planet mass, a_p the planet semi-major axis, and P_p the planet's orbital period. The planet-to-total mass ratio is $\mu \equiv M_p/(M_* + M_p)$. In addition, we choose $G = 1$, such that the star and the planet masses are $1 - \mu$ and μ respectively. This choice of units reduces the relevant physical parameters of our simulations to two: the mass ratio μ and the shearing viscosity ν.

Surface Density Profile. Typically, the surface density in circumstellar discs is modeled with a power-law profile (i.e. $\Sigma(R) = \Sigma_0(R/R_0)^{-p}$). Following de Val-Borro et al. (2006) we choose our discs to have constant surface density ($p = 0$) of $\Sigma = \Sigma_0 = 0.02M_*/(\pi a_p^2)$, such that the enclosed mass at the planet position is $\sim 2\%$ of the mass of the star (the disc is assumed to extend all the way to the star, beyond the inner boundary). Note that real discs have steeper density profiles, $-3/2 < p < -1$ (Andrews et al., 2009).

Planet Mass. In the units used here, the planet and star masses are determined by a single parameter μ. We have explored two values of μ: 10^{-3} and 10^{-4} (Table 6.1). For a stellar mass of $M_\odot$, these mass ratios correspond to the planet masses of Jupiter and Saturn, respectively. Since simulations have been shown to develop numerical artifacts if planets are added impulsively into the disc, we increase the planet mass from zero up to M_p slowly in time as (de Val-Borro et al., 2006):

$$M_p(t) = \begin{cases} M_p \sin^2\left(\frac{\pi t}{10P_p}\right) & t \leq 5P_p \\ M_p & t > 5P_p \end{cases} \tag{4.5}$$

Temperature Structure. For simplicity, we use a globally isothermal equation of state in all simulations unless stated otherwise. We use an exact (iterative) isothermal Riemann solver (e.g,) This implies that the disk aspect ratio ($h \sim c_s/v_k$) is not constant. The

Orbital Velocity Profile. Our disc simulations are started from centrifugal equilibrium and axial symmetry. From the Euler equations in cylindrical coordinates,

the equilibrium and axisymmetry condition implies an azimuthal velocity that satisfies

$$\frac{v_\phi^2}{R} = \frac{\partial \Phi}{\partial R} + \frac{1}{\Sigma}\frac{\partial P}{\partial R} \quad ,$$

where the Keplerian term $R(\partial\Phi/\partial R) = v_K^2 = \sqrt{GM_*/R}$ is modified by a "pressure buffer" term, resulting in an orbital velocity that is slightly sub-Keplerian (for $c_s^2 \ll v_K^2$)

$$v_\phi^2 = v_K^2 - c_s^2(l+p) \quad , \tag{4.6}$$

where the pressure buffer term comes from the initial temperature and density gradients: $\partial P/\partial R = (\partial P/\partial c_s^2)(dc_s^2/dR) + (\partial P/\partial \Sigma)(d\Sigma/dR)$. Since in this work we use an initially constant surfance density and a globally isothermal equation of state, $p = l = 0$, so the initial rotation curve is strictly Keplerian.

Initial Mesh. The setup of the initial mesh is entirely arbitrary, and can be chosen according to the needs of the problem being simulated. Figure 4.4 shows a disk mesh constructed from an initial polar distribution of mesh-generating points. After one orbit, the mesh alignment has been removed, but the distribution of cells in the radial direction is nearly preserved. Figure 4.5 shows a similar setup (the number of cells in the innermost ring is the same as in Figure 4.4) but now the initial placement of the mesh-generating points follows a cartesian grid. After one orbit, the mesh has relaxed, and cells are roughly concentrated along different concentric rings. Polar-like meshes have constant azimuthal resolution at all radii, which is useful for comparing to polar-grid codes. However, cartesian like meshes preserve a near-constant cell size and cell aspect ratio, which guarantees each cell to have a roughly constant number of neighbors and not cell size asymmetry as in the polar-like case. Unfortunately, the cartesian grid with linear reconstruction cannot capture well the axisymmetric motion of the fluid, and

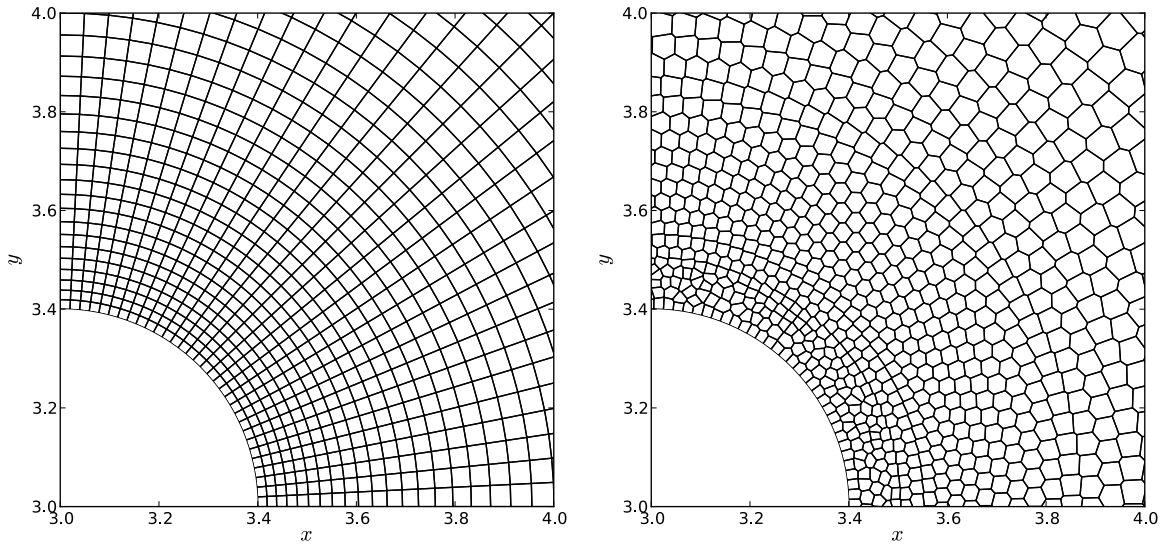
errors are seeded at the onset, regardless of the near axisymmetric distribution of cells that can be reached at a later time. A third alternative, and the one we adopt as default in our simulations, is to position the cells from the beginning as in the right panel of Figure 4.4, i.e., to setup the initial distribution of mesh-generating points by creating a set of concentric rings of constant separation in the radial direction. The same separation is imposed between points in the azimuthal direction, thus keeping the aspect ratio and cell size nearly constant throughout the computational domain.

4.3 Results

4.3.1 Surface Density Field

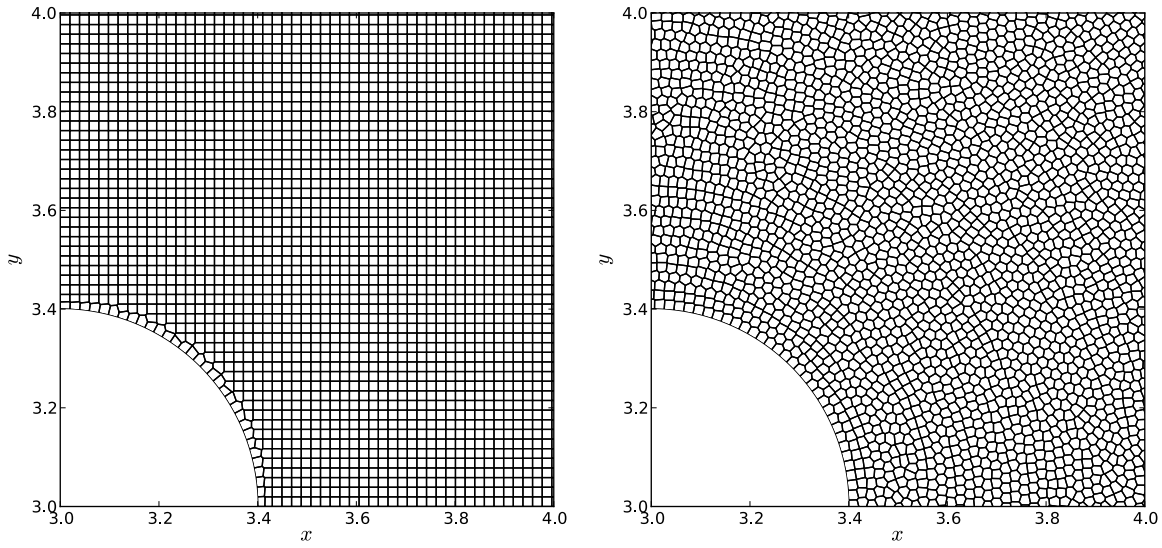
The perturbed surface density profile of the disc provides a qualitative means to asses the relative performance of different numerical schemes. The key features on which we base our comparison are the overall shape of the tidal wake launched by the planet, its location with respect to the predictions of linear theory (Ogilvie & Lubow, 2002), and how far from the planet the wake is damped. For the Jupiter-mass simulations, the shape of the gap carved by the local deposition of angular momentum is an important diagnostic for the accuracy and numerical diffusivity of the code. In the code comparison of de Val-Borro et al. (2006), the different numerical schemes differed on the sharpness of the gap, its degree of axisymmetry, its depth, the smoothness of the remaining material, and the amount of gas retained around the orbit's Lagrangian points after 100 or more orbits.

Figure 4.6 shows the surface density field of the two inviscid fiducial simulations



(a) (b)

Figure 4.4.— Low resolution example of a quasi-regular polar mesh. Left panel: polar grid with logarithmic spacing in r and uniform spacing in ϕ (128 azimuthal zones) that guarantees a constant aspect ratio. The mesh is approximately polar, since all intercell boundaries are straight lines and do not take any curvature into account. Right panel: mesh topology at time $t = 2\pi$ (one orbit). In this case, the rough axisymmetry of the mesh is preserved, due to the azimuthal motion of the gas, that does not disturb the mesh-generating points in the radial direction. In this case, the mesh regularization technique the cells per logarithmic interval in radius, maintaining the angular resolution (128 azimuthal zones) at all radii.



(a) (b)

Figure 4.5.— Similar to Figure 4.4 but with an initially cartesian distribution of mesh-generating points. After the mesh has relaxed, the cells are roughly arranged along concentric rings of similar width.

Numerical experiment (label/radial zones)	Physical parameters μ	ν	Effective azimuthal resolution $N_\phi(R_{\rm in})$	$N_\phi(R_p)$	$N_\phi(R_{\rm out})$
JUP_128	10^{-3}	–	~ 90	~ 360	~ 890
JUP-VISC-A_128	10^{-3}	10^{-4}	”	”	”
JUP-VISC-B_128	10^{-3}	10^{-5}	”	”	”
JUP-VISC-C_128	10^{-3}	10^{-6}	”	”	”
JUP_256	10^{-3}	–	~ 180	~ 720	~ 1780
JUP-VISC-A_256	10^{-3}	10^{-4}	”	”	”
JUP-VISC-B_256	10^{-3}	10^{-5}	”	”	”
JUP-VISC-C_256	10^{-3}	10^{-6}	”	”	”
JUP_512	10^{-3}	–	~ 360	~ 1440	~ 3560
JUP-VISC-A_512	10^{-3}	10^{-4}	”	”	”
JUP-VISC-B_512	10^{-3}	10^{-5}	”	”	”
JUP-VISC-C_512	10^{-3}	10^{-6}	”	”	”
NEP_128	10^{-4}	–	~ 90	~ 360	~ 890
NEP-VISC-A_128	10^{-4}	10^{-4}	”	”	”
NEP-VISC-B_128	10^{-4}	10^{-5}	”	”	”
NEP-VISC-C_128	10^{-4}	10^{-6}	”	”	”
NEP_256	10^{-4}	–	~ 180	~ 720	~ 1780
NEP-VISC-A_256	10^{-4}	10^{-4}	”	”	”
NEP-VISC-B_256	10^{-4}	10^{-5}	”	”	”
NEP-VISC-C_256	10^{-4}	10^{-6}	”	”	”
NEP_512	10^{-4}	–	~ 360	~ 1440	~ 3560
NEP-VISC-A_512	10^{-4}	10^{-4}	”	”	”
NEP-VISC-B_512	10^{-4}	10^{-5}	”	”	”
NEP-VISC-C_512	10^{-4}	10^{-6}	”	”	”

Table 4.1: Simulation parameters. For each planet mass ($\mu = 10^{-3}$ and $\mu = 10^{-4}$) we vary resolution (in terms of the number of radial zones) and the viscosity coefficient. The different resolutions are $N_R = 128$, 256 and 512 radial zones. The different viscosities are $\nu = 10^{-4}$, 10^{-5} and 10^{-4} as well as runs not including explicit viscous terms. The bold type-face denote the base of “fiducial” runs: an inviscid and a viscous ($\nu = 10^{-5}$) run for each mass ratio at a resolution of $N_R = 128$.

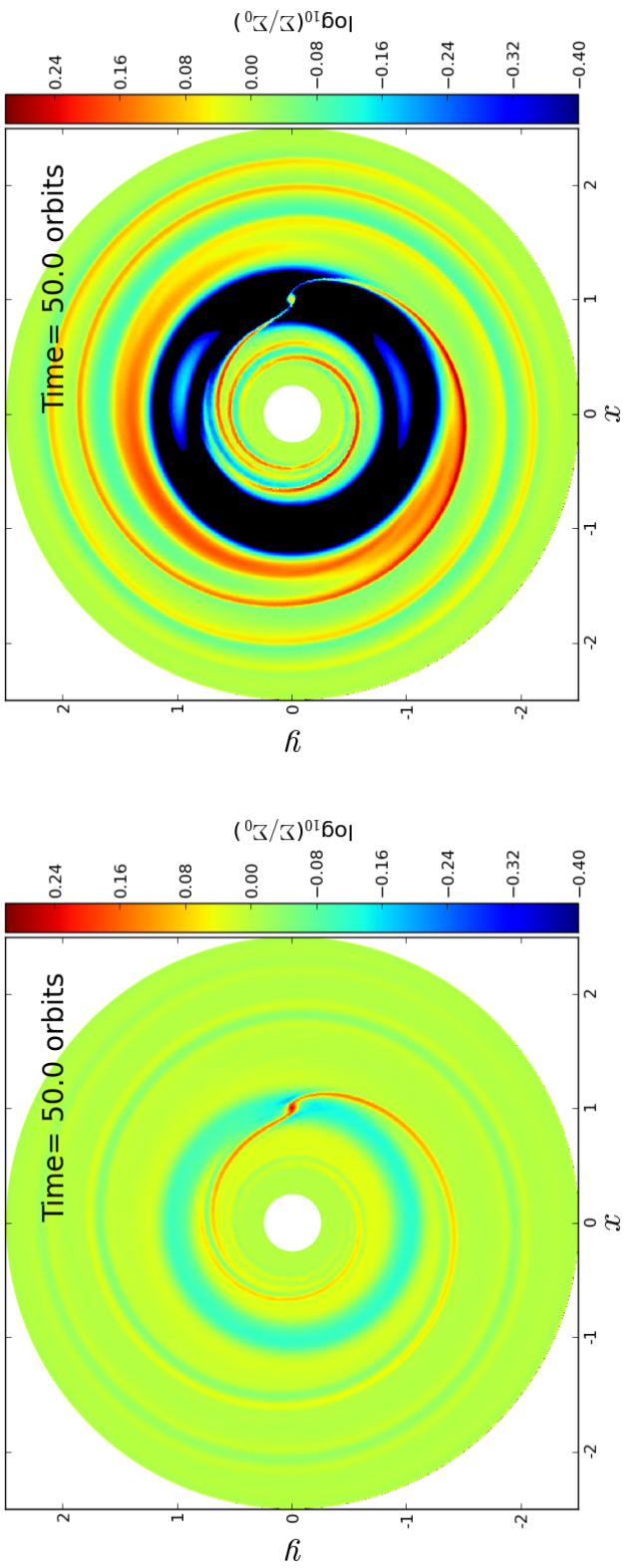


Figure 4.6.— Inviscid fiducial simulations (128 radial zones) for a Neptune-mass planet (left panel; $\mu = 10^{-4}$) and a Jupiter-mass planet (right panel; $\mu = 10^{-3}$) after 50 orbits. The color scale is set to be identical for both panels ($-0.4 < \log_{10}(\Sigma/\Sigma_0) < 0.3$) to highlight the difference in gap width and depth between the Neptune-mass and Jupiter-mass cases.

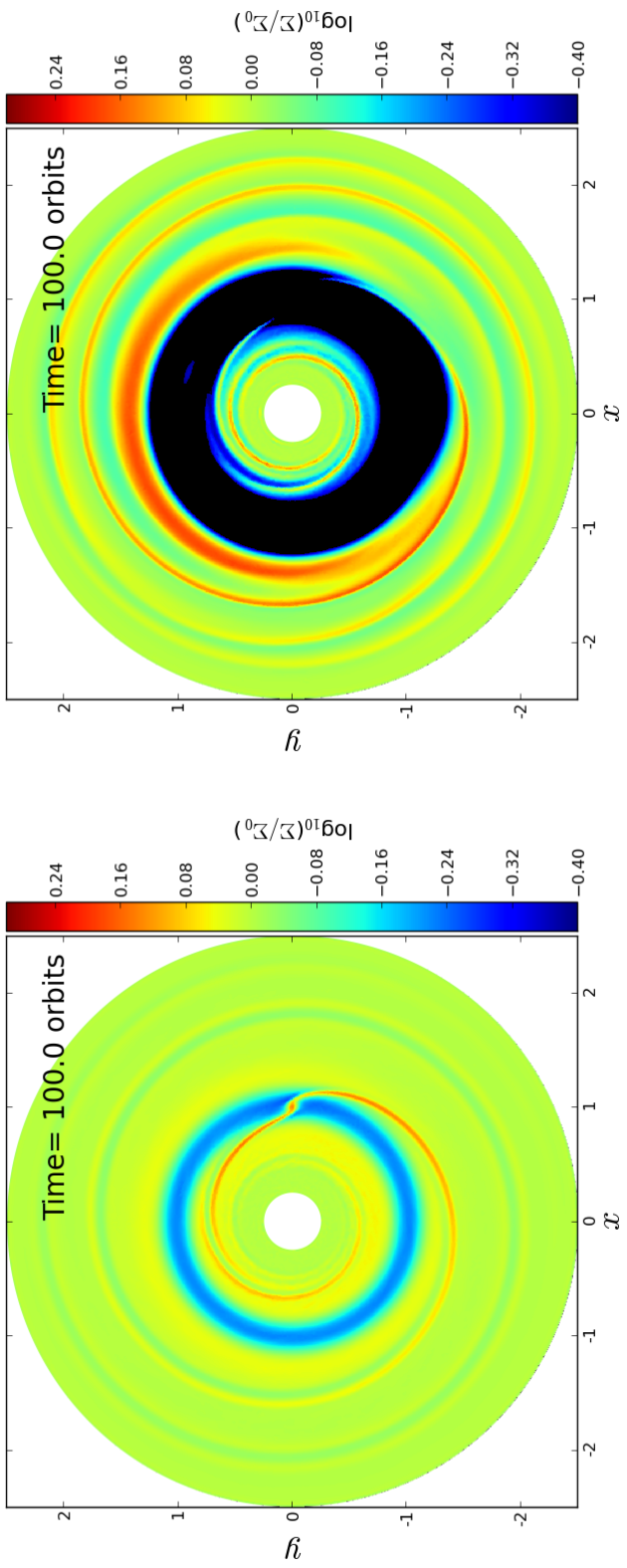


Figure 4.7.— Inviscid fiducial simulations (128 radial zones) for a Neptune-mass planet (left panel; $\mu = 10^{-4}$) and a Jupiter-mass planet (right panel; $\mu = 10^{-3}$) after 100 orbits.

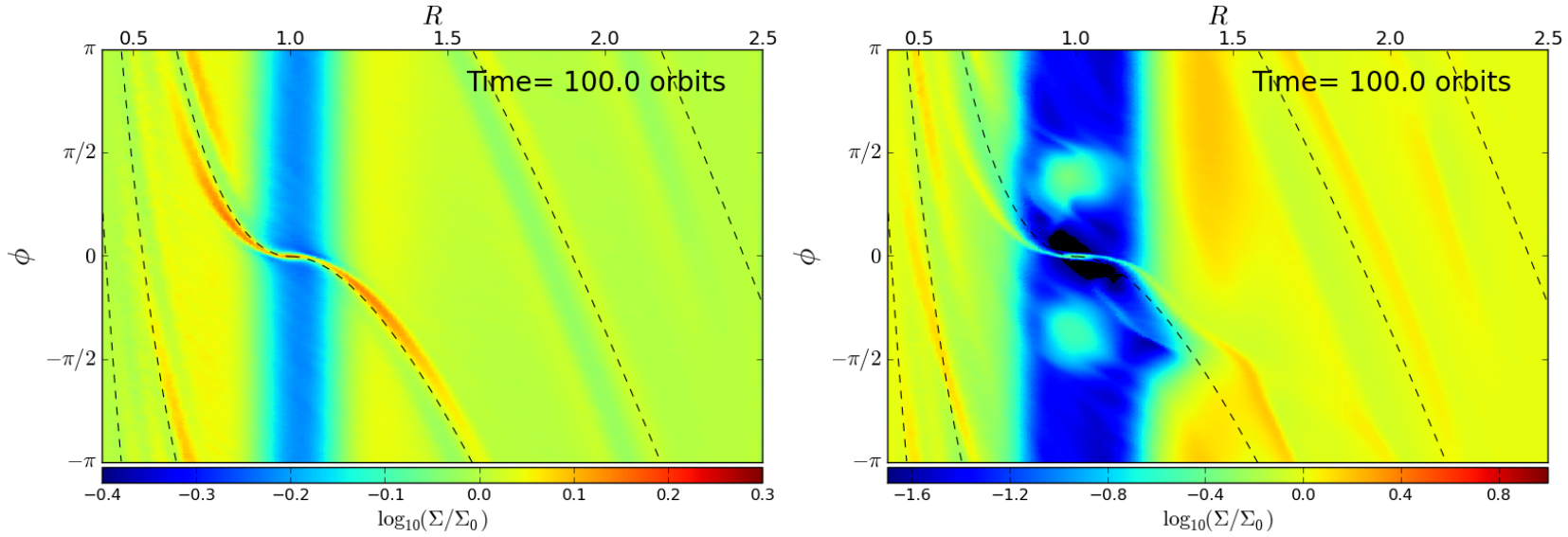


Figure 4.8.— Same data as in Figure 4.7 but displayed in polar coordinates. The color scale range has been chosen to match the ones used by de Val-Borro et al. (2006); for the Neptune-mass case (left panel), the color-coded density range is $-0.4 < \log_{10}(\Sigma/\Sigma_0) < 0.3$, while for the Jupiter-mass case (right panel), it is $-1.7 < \log_{10}(\Sigma/\Sigma_0) < 1.0$. In addition, a similar color table has been chosen for easier comparison. Inviscid fiducial simulations (128 radial zones) for a Neptune-mass planet (left panel; $\mu = 10^{-4}$) and a Jupiter-mass planet (right panel; $\mu = 10^{-3}$) after 100 orbits. The dashed lines illustrate the location of the wake in the linear regime.

(128 radial zones) for mass ratios of $\mu = 10^{-4}$ (Neptune-mass planet) and $\mu = 10^{-3}$ (Jupiter-mass planet) after 50 orbits. Although the disk is globally isothermal, and therefore the precise locations in which the planet is torquing the gas distribution differ from the case with a radial temperature gradient, qualitatively, we find agreement with the known results in the literature. Roughly speaking, according to the gap-opening criterion of Lin & Papaloizou (1993), Jupiter-mass planets are expected to clear gaps in a few tens of orbits, Neptune-mass planets are not. According to Lin & Papaloizou (1993), a planet of mass M_p will open a gap in the disk if

$$M_p > M_{\rm th} \equiv \frac{c_{s,p}^3}{\Omega_p G} \quad . \tag{4.7}$$

For the parameters used in the present work, the so-called thermal mass $M_{\rm th}$ is $\sim 1.25 \times 10^{-4}$, which is slightly above the Neptune-mass planet ($M_p = 10^{-4}$) and eight times smaller than the Jupiter-mass planet ($M_p = 10^{-3}$). For further discussion of gap-opening and its dependence on resolution and viscosity coefficient, see Section 4.3.2 below.

For easier comparison with earlier work, we have applied a coordinate transformation to the cell coordinates and converted the density field from the x-y plane into the R-ϕ plane. In Figure 4.8 we show the two fiducial inviscid simulations in polar coordinates, with color scale limits matching those of de Val-Borro et al. (2006) and using a similar color table. In addition, Figure 4.8 includes the shape of the planetary wake derived in

the linear regime by Ogilvie & Lubow (2002) [1] .The tidal wake excited by a planet in a disk is a result of sound waves excited by a perturber moving at supersonic speeds and it arises as the superposition of all harmonics In the linear regime, the wake profile asymptotically reaches a stationary profile (Narayan et al., 1987; Rafikov & Petrovich, 2012) as a result of and its is proportional to the planet mass (e.g. Dong et al., 2011b; Rafikov & Petrovich, 2012). Like sound waves, these linear waves are non-dispersive. However, as their amplitude grows, and unless diffusion mechanisms (either physical or numerical) damp them first, these wakes inevitably steepen and become double-valued (Goodman & Rafikov, 2001; Rafikov, 2002). For large enough planets, the wake steepens into a shock close to the planet, depositing angular momentum locally (Goodman & Rafikov, 2001; Rafikov, 2002). This process is ultimately what causes the opening of a gap in the vicinity of the planet (Lin & Papaloizou, 1986b). Once there is a gap present, the gravitational coupling between the disk and the planet cannot be calculated in the linear regime assuming a uniform background density, since this will lead to erroneous estimates of the angular momentum flux (Petrovich & Rafikov, 2012). Not only does the torque in gapped systems differ from the original calculation of Goldreich & Tremaine (1979) but also the shape of the tidal wake differs. While in the Neptune-mass simulations the gap is not deep enough to significantly alter the shape of the wake compared to the theoretical results in the linear regime (Figure 4.8, left panel), the

[1]The location of the planetary wake, for $R_p = 1$, is given by

$$\phi(R) = \begin{cases} 2\pi(t/P_p) - \dfrac{2}{3}h^{-1}\left(R^{3/2} - \dfrac{3}{2}\ln R - 1\right) + \pi & R < 1 \\ 2\pi(t/P_p) + \dfrac{2}{3}h^{-1}\left(R^{3/2} - \dfrac{3}{2}\ln R - 1\right) + \pi & R > 1 \end{cases} .$$

with $h = h_p(R/R_p)^{1/2}$, where $h_p = 0.05$.

Jupiter-mass runs (Figure 4.8, right panel) show an evident mismatch between the theoretical wake position and the actual density maxima of the wake.

Note that in the globally isothermal case, the location of the Lindblad resonances are different from those in a disk with a radial temperature gradient; in the present case, the spiral wake is less tightly wound than in the setup of de Val-Borro et al. (2006), which uses a $T \propto R^{-1}$ temperature profile.

The surface density field in polar coordinates emphasizes the departure from axial symmetry of the density gap cleared by the Jupiter-mass planet (Figure 4.8, right panel). This is something to be aware of in moving-mesh codes, which are known to develop morphological asymmetries faster than fixed grid codes (c.f. Springel, 2010a). The early development of asymmetries during the growth of instabilities can be caused by *general* features of grid codes, not exclusive to moving-mesh codes. The main known cause of the numerical seeding of asymmetries in initially symmetric initial conditions is the not-so-often-discussed "grid noise". The location of cell centers as well as of face centers are also subject to roundoff error, ultimately affecting the evolution of the fluid quantities. In moving-mesh codes, grid noise can have an even greater relevance, since it is not only determined by round-offs, but also by truncation error in the mesh-drifting algorithm, since the location of the mesh-generating points is evolved to finite accuracy.

AREPO updates the mesh-generating points using a leapfrog integrator, which should amplify the noise in the grid compared to static grid codes. Sophisticated methods to limit grid noise employ structured and symmetrized methods for evolving the different dimensions in multi-dimensional flows. Codes than implement "symmetric sweeping" of the grid faces to preserve the symmetries of the flow to greater accuracy (cf. the examples

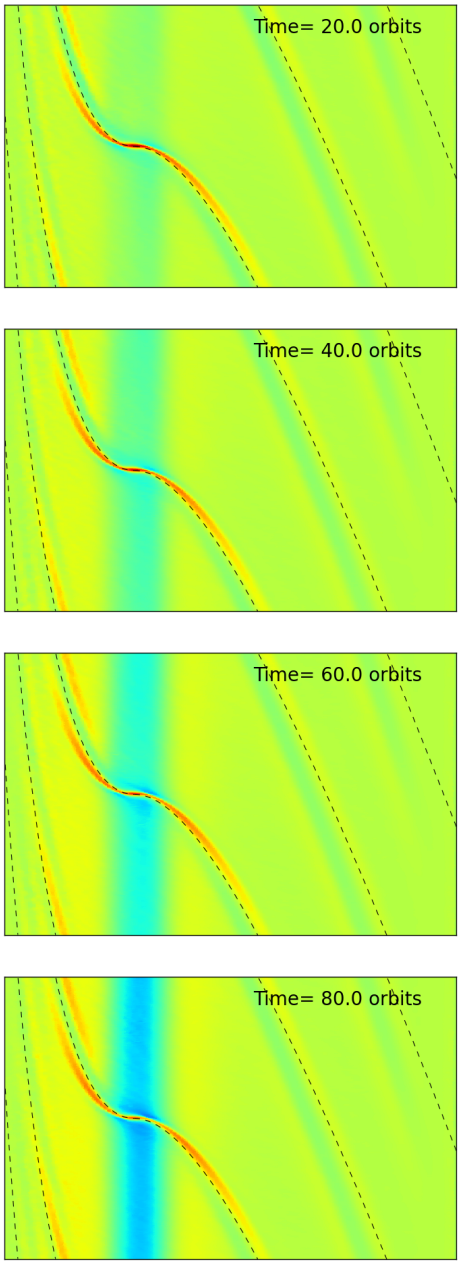


Figure 4.9.— Surface density field for the fiducial inviscid Neptune-mass simulation at four different times: 20, 40, 60 and 80 orbits. The coordinate axes and color key (not shown for clarity) are identical to those of Figure 4.8 (left panel). The black dashed line corresponds to the theoretical position of the tidal wake in the linear regime and assuming a uniform background density. The density field outside the planet's coorbital region is nearly stationary for several tens of orbits.

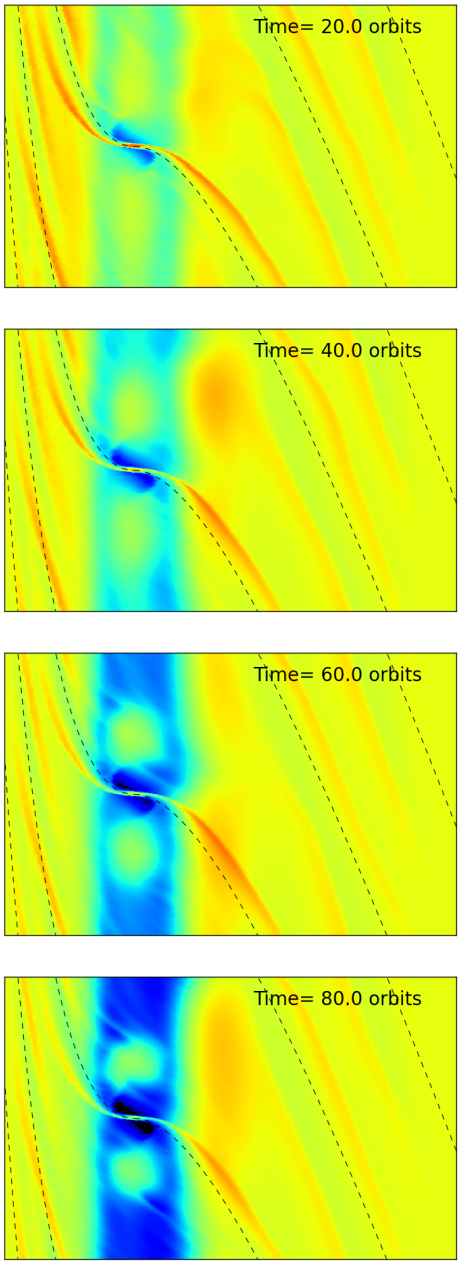


Figure 4.10.— Similar to Figure 4.9, but for a Jupiter-mass planet. The coordinate axes and color key (not shown for clarity) are identical to those of Figure 4.8 (right panel). The black dashed line corresponds to the theoretical position of the tidal wake in the linear regime and assuming a uniform background density. In contrast with the Neptune-mass case (Figure 4.9), the density field shows variability over long ($\sim$ tens of orbits) and short ($\lesssim 1$ orbit) timescales.

of oblique 2-D shock tubes in ATHENA; see also Sijacki et al., 2012). For discussions on symmetric sweeping we refer the reader to Aloy et al. (1999) and Sutherland (2010). Minimizing grid noise by mechanisms analogous to symmetric sweeping does not seem feasible for unstructured moving meshes, in which there is no ordered list of cell faces nor a constant number thereof.

Although the amplification of asymmetries in the flow can be favored by a moving-mesh, those are *not generated* by the mesh itself in our planet-disk interaction simulations. To test this, we compare the density field in our fiducial inviscid simulations (`NEP_128` and `JUP_128`) at different times, looking for transients. In Figure 4.8, the density field of the Neptune-mass simulations is shown for four different times (20, 40, 60 and 80 orbits). Aside from a gradually carved shallow gap, the background density is nearly stationary over a period of 100 orbits, with no sign of instabilities or transient over-densities. Figure 4.10 shows the density field at 20, 40, 60 and 80 orbits for a Jupiter-mass planet In this case, the density field outside the gap shows variations that can be of the order of the density peaks associated to the spiral wake.The nearly stationary density field developed by the fiducial Neptune-mass run in the planet's rotating frame shows that the development of strong density bumps and vortices outside the gap in the Jupiter-mass case is not a feature of AREPO but a consequence of the large gap carved by the planet.

We conclude that the sensitivity to the development of asymmetries does not arise spontaneously in AREPO, but only in hydrodynamicaly unstable regimes. The importance of the preservation of symmetries (which do not occur in nature) well into the non-linear regime in an unstable configuration is a matter of debate among researchers (e.g., McNally et al., 2012a). Although we do not believe it is a critical component of

a hydrodynamical scheme, future work should address pathological cases in which grid noise might affect the convergence rate in a code like AREPO.

The transient features observed in Figure 4.10 but entirely absent in Figure 4.9 correspond to the "edge instabilities" observed previously in numerical simulations by Koller et al. (2003),Li et al. (2005) and de Val-Borro et al. (2006). These edge effects are usually associated with vortices right outside the gap (de Val-Borro et al., 2007; Lyra et al., 2009) as can be corroborated by the absolute vorticity field shown in Figure 4.11 (right panel), which shows a local vorticity maxima with the greatest deviation from axisymmetry at the gap edge (see Figure 4.8, right panel). In contrast, the perturbations to the disk vorticity field exerted by a Neptune-mass planet at the fiducial resolution are very small (Figure 4.11, left panel), which is consistent with the lack of transients observed in the density field in the same simulation (Figure 4.9).

The formation of large vortices in disk simulations can have large effects on the evolution of planets. On one hand, the vortices themselves can exert bursty or oscillating torques on the planet on timescales of order the orbital time. For example, in the numerical study of Li et al. (2005), the authors argue that this oscillating nature of torques can reverse the direction of planet migration. On the other hand, vortices themselves favor the planet-forming process, since anticyclonic vortices in disks have been found to favor the concentration of dust particles (e.g. Barge & Sommeria, 1995). We will explore the vorticity structure of disks with gaps in Section 4.3.3.

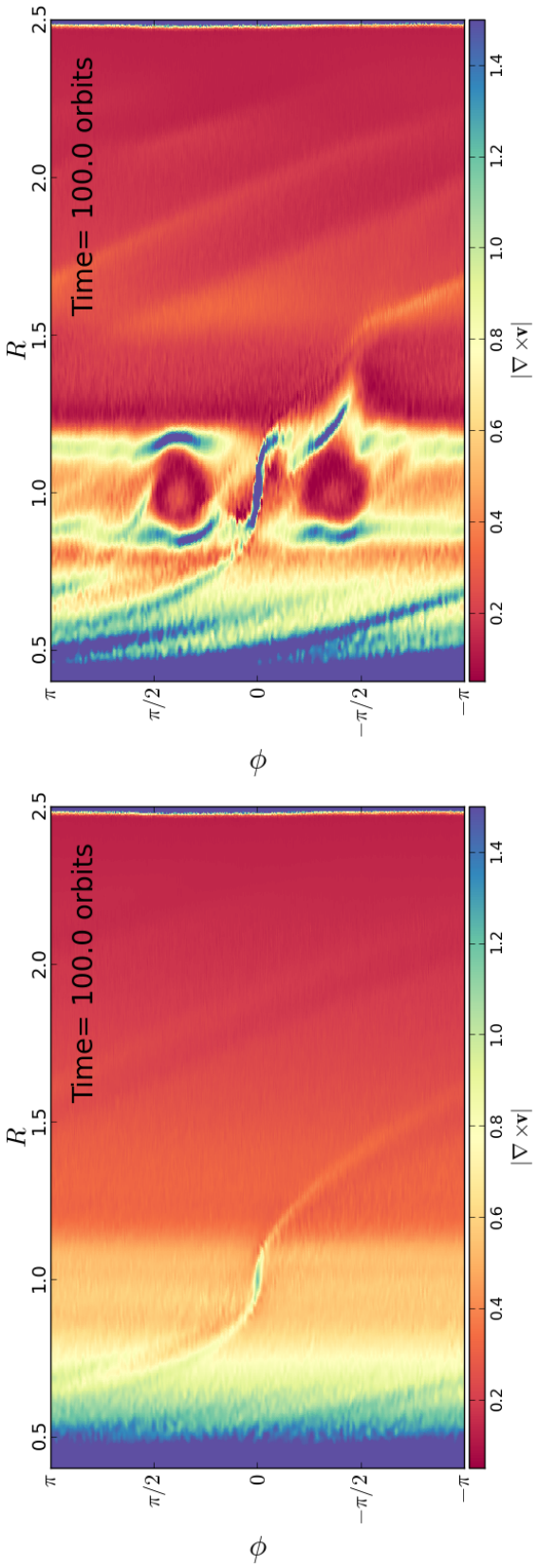


Figure 4.11.— Absolute vorticity field for the inviscid fiducial simulations with $\mu = 10^{-4}$ (left panel) and $\mu = 10^{-3}$ (right panel).

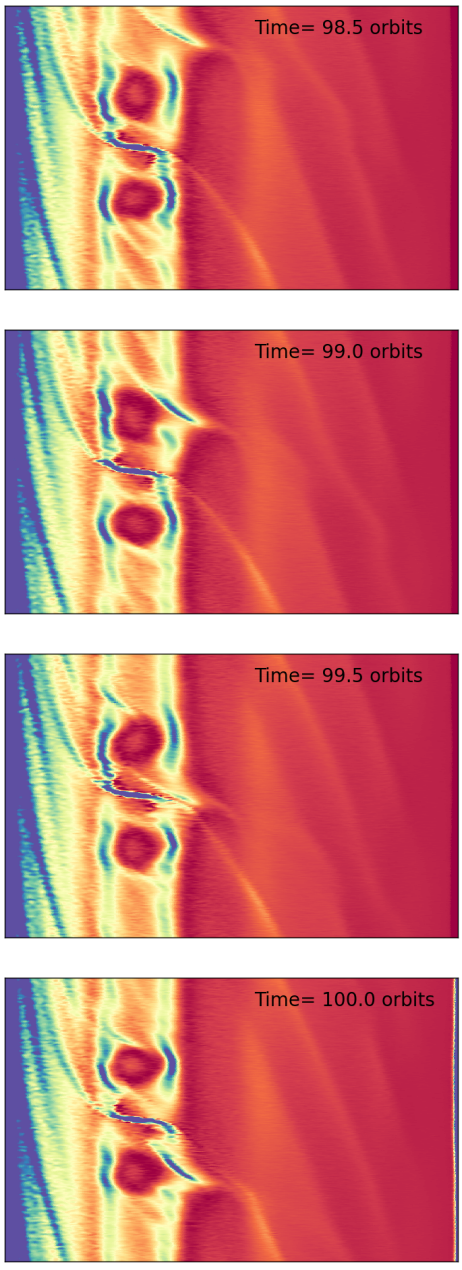


Figure 4.12.— Vorticity evolution for the Jupiter-mass simulations over short timescales.

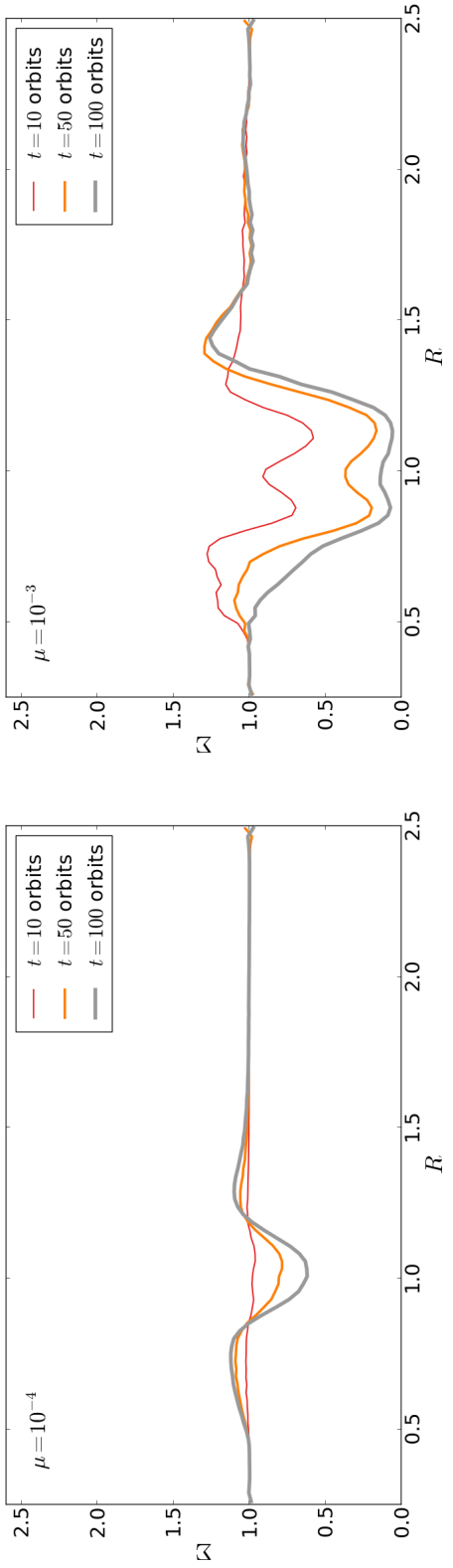


Figure 4.13.— Top panel: azimuthally averaged surface density profile for the inviscid fiducial simulation with $\mu = 10^{-4}$ at three different times (10, 50 and 100 orbits). Bottom panel: same as top panel but with $\mu = 10^{-3}$.

4.3.2 Surface Density Profile and Gap Opening

Figure 4.13 shows the evolution of the (azimuthally averaged) surface density profile for the Neptune-mass (top panel) and Jupiter-mass (bottom panel) planets, showing how a shallow gap is carved in the former case, and a deep gap in the latter. In the Neptune-mass case, the density field shows a dip within the coorbital region of about 40% of its initial value. This deficit increases to over 90% in the Jupiter-mass case after 100 orbits. Outside the gap, the background flow in the Neptune-mass case is solely disturbed by the spiral wake launched by the planet at the Lindblad resonances; the averaged density field, however, is barely disturbed. The Jupiter-mass planet, on the other hand, does alter the background mean flow significantly, in part owing to the larger amplitude of the spiral wake (in the linear regime, the amplitude of the wake is proportional to M_p) and to the density bump caused by material being displaced from the coorbital region. Indeed, a significant part of the mass in the coorbital region is relocated right outside the gap edge. This density increase can be very important for the development of vortices in the disk (e.g., Lovelace et al., 1999; see Section 4.3.3). This flow of mass out of the coorbital region can, in principle, go into the wave absorbing region, which will damp out the overdensity, which will violate mass conservation and translate into a mass loss rate (see de Val-Borro et al., 2006). Since mass loss seems to occur in our Jupiter-mass run, which shows a density bump both at the interior and exterior edges of the gap at early times, but only a density bump at the exterior edge at later times. Higher-resolution runs show that a density maximum should also form close to the edge of the wave-absorbing region.

These results are in agreement with the "thermal criterion" of Equation 4.7, which

state that our Jupiter-mass simulations should open a gap while the Neptune-mass runs should not. However, the thermal criterion has been questioned by recent evidence of planets with masses smaller than the thermal mass being able to open gaps (e.g. Dong et al., 2011a; Duffell & MacFadyen, 2012; Zhu et al., 2013). An indication of this can be seen in Figure 4.14, where upon increasing the resolution of the inviscid Neptune run (top row, left-to-right), the gap carved by the planet is deeper and sharper at the edges.

The objections to the thermal criterion were raised by Goodman & Rafikov (2001) (see also Rafikov, 2002), whose calculations showed that the local deposition of angular momentum from the launched waves onto the disk that causes the clearing of the gap happens more efficiently if these waves shock at a finite distance from the planet $x_{\rm sh}$:

$$|x_{\rm sh}| \approx 0.93 \left(\frac{\gamma+1}{12/5} \frac{M_p}{M_{\rm th}} \right)^{-2/5} H_p \ . \tag{4.8}$$

Evidently, wave steepening depends on grid resolution, thus for inviscid simulations, the gap shape and depth will depend on whether the angular momentum is injected into the disk by shock dissipation or by a viscous-like dissipation onto the computational grid.

For the isothermal ($\gamma = 1$) examples shown here, and recalling that the planet mass in the Neptune case is $M_p = 4/5 M_{\rm th}$, the shock distance given by Equation 4.8 is $|x_{\rm sh}| \approx 1.09 H_p \sim 0.055$. For the Jupiter case ($M_p = 8 M_{\rm th}$), we have $|x_{\rm sh}| \approx 0.44 H_p \sim 0.022$. Note that, in the vicinity of the planet, the radial extent of a Voronoi cell at $t = 0$ is 1.8×10^{-2}, 9.0×10^{-3} and 4.5×10^{-3}, for $N_R = 128, 256$ and 512 respectively. Therefore, the shock distance is not resolved whatsoever for the lowest resolution runs. Furthermore, the shock distance for the Jupiter-mass case is smaller than the gravitational softening used in our simulations (as in those by de Val-Borro et al., 2006), thus the precise gap-opening mechanism cannot be adequately captured

in these simulations. We remind the reader that de Val-Borro et al. (2006) chose the softening parameter of $\varepsilon = 0.06 H_p$ based on the mimicking of the saturation of the torques in three dimensions that this choice provides.

Viscosity (or viscous-like numerical diffusivity) not only affects the mechanism by which gaps form, but also affects the long term balance between the tidal torque and the viscous torque (Crida et al., 2006, 2007). Balancing the angular momentum flux of these two competing effects, Zhu et al. (2013) find a "viscous criterion" of the form

$$\left(\frac{M_p}{M_*}\right)^2 \gtrsim \frac{10\nu H_p^3}{R_p^5 \Omega_p} \quad , \tag{4.9}$$

which, for the disk parameters of this work, reduces to $M_p/M_* \gtrsim \sqrt{1.25 \times 10^{-3}\nu}$. This inequality means that the critical mass ratio $\mu \approx M_p/M_*$ should be above $\sim 1.11 \times 10^{-4}$ for a viscosity coefficient of $\nu = 10^{-5}$, i.e., the Neptune-mass case is again border-line capable of opening a gap. For the other viscosity coefficients we have experimented with in this work, 10^{-4} and 10^{-6}, we find that M_p/M_* must be greater than 3.5×10^{-4} and 3.5×10^{-5} respectively. Therefore, with $\nu = 10^{-4}$, gap formation should be suppressed for the Neptune-mass planet and significantly affected for the Jupiter-mass planet. With $\nu = 10^{-6}$, both the Neptune-mass and the Jupiter-mass planets should be able to open partial or full gaps, provided numerical diffusion is less important than the physical viscosity (Figure 4.14 and Figure 4.15).

Unless physical viscosity is included in this kind of problem, the result of numerical simulations should be expected to converge slowly. If the clearing of a gap is indeed determined by local deposition of angular momentum due to the shock dissipation of waves, very high resolution is needed to properly capture the shape of the tidal wake and its steepening(cf. Dong et al., 2011b,a). If the steepening of the wake is

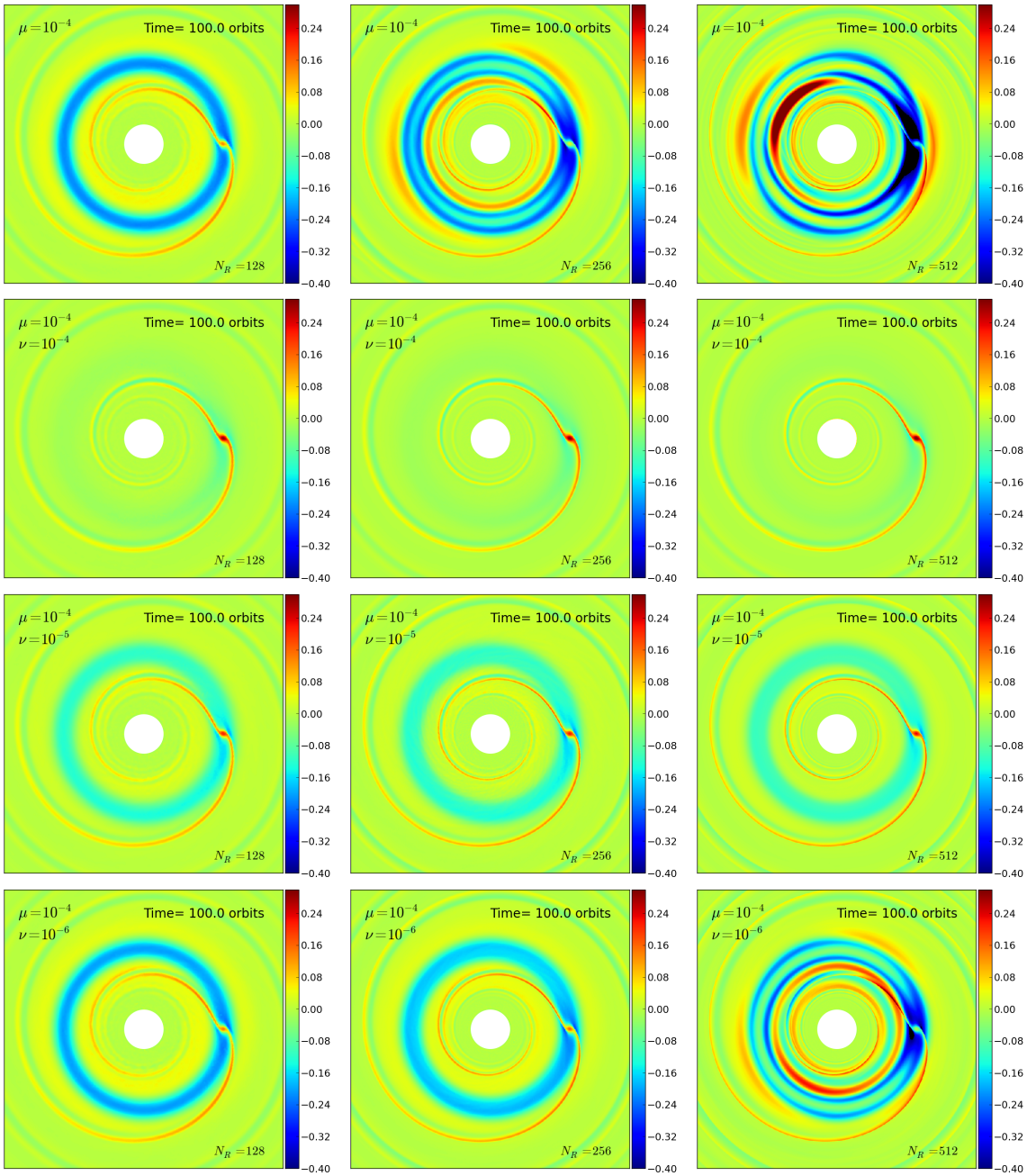


Figure 4.14.— Surface density fields for Neptune-mass ($\mu = 10^{-4}$)simulation with different resolutions (for different columns) and physical viscosity coefficients (for different rows). First, second and third column have 128, 256 and 512 radial zones respectively. First row corresponds to inviscid (i.e., no explicit viscosity terms) runs; second, third and fourth row have viscosity coefficients of $\nu = 10^{-4}$, 10^{-5} and 10^{-6} respectively.

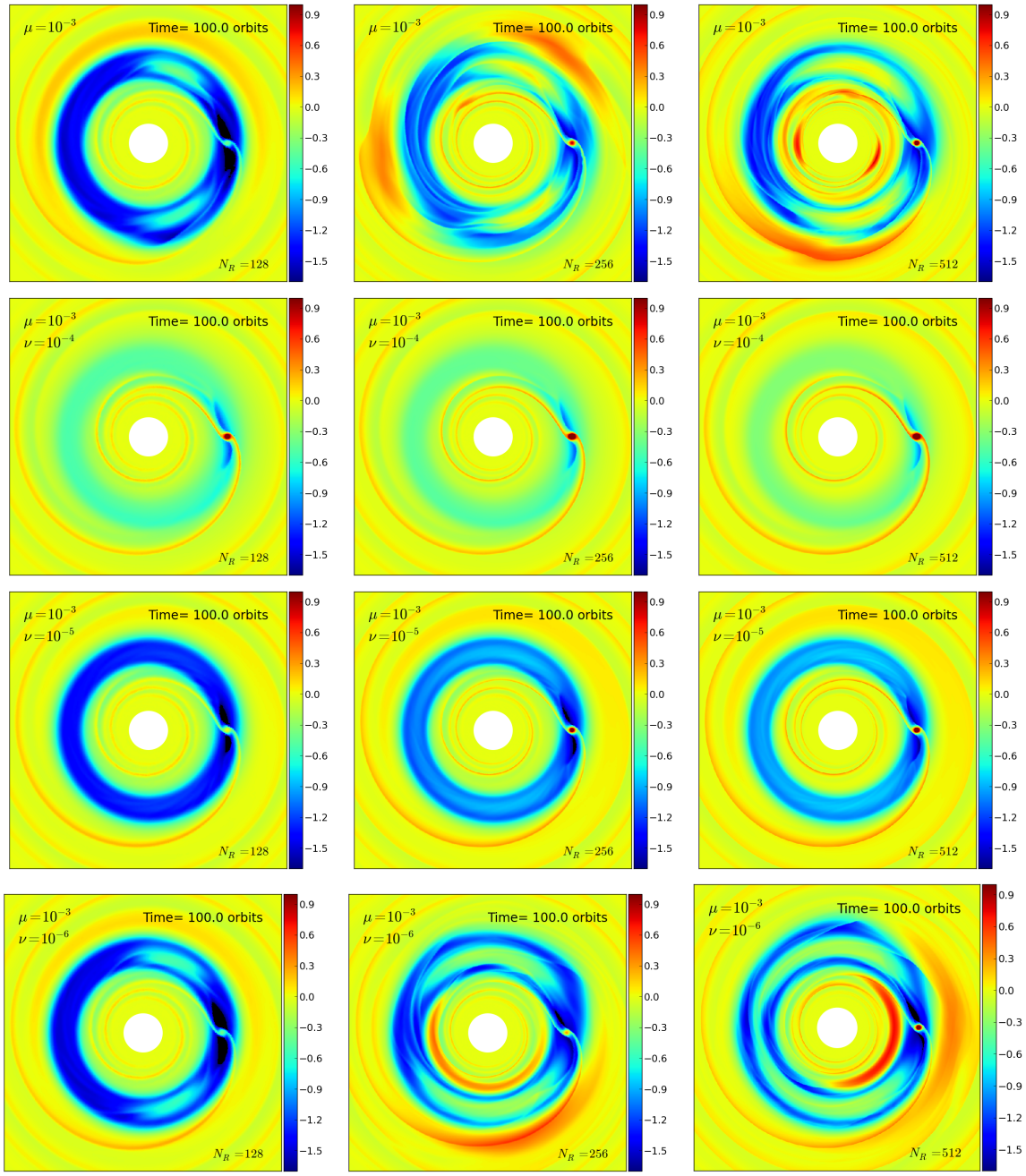


Figure 4.15.— Same as Figure 4.14 but for a Jupiter-mass planet ($\mu = 10^{-3}$).

not resolved, the deposition of angular momentum in the coorbital region is likely to be dominated by diffusion onto the grid, thus mimicking the linear damping of waves described by Takeuchi et al. (1996). In Figures 4.14 and 4.15, the shape of the gap shows a qualitatively rapid convergence for the viscous simulations. The inviscid simulations, in contrast, show no converging trend. In addition, a small enough softening parameter is needed, such that the shock takes place at a distance from the planet where the gravitational potential due to it is truly Newtonian (Dong et al., 2011b,a).

Figure 4.16 shows the azimuthally-averaged surface density field for the inviscid (top panel) and the viscous (bottom panel) Neptune simulations after 100 orbits for three different resolutions: $N_R = 128$, 256 and 512. The surface density profiles include a 1-σ contour to quantify the degree of axisymmetry in the disk. As suspected already from the two-dimensional density contours (Figure 4.14), there is no convergence in the inviscid simulations, while very good consistency is found in the viscous runs across the different resolutions, in addition to uniformity in the azimuthal direction. Interestingly, the greatest deviation in the inviscid runs comes from the region of the disk interior to the planet, a possible indication that the azimuthal resolution of the fiducial simulations in that region is too poor to capture the density evolution adequately (recall that our runs have a varying number of azimuthal zones with radius).

Similarly, Figure 4.17 shows the azimuthally-averaged surface density profile for the Jupiter-mass simulations. In the inviscid case, the edge instabilities cause much wilder variations than in the Neptune analog. As in the inviscid Neptune-mass case, the inviscid Jupiter-mass runs show greater variability in the inner region as resolution is increased. In the viscous case, the flow is smoother and deviations from the azimuthal mean are smaller, although the 1-σ contours still show broader scatter than the Neptune

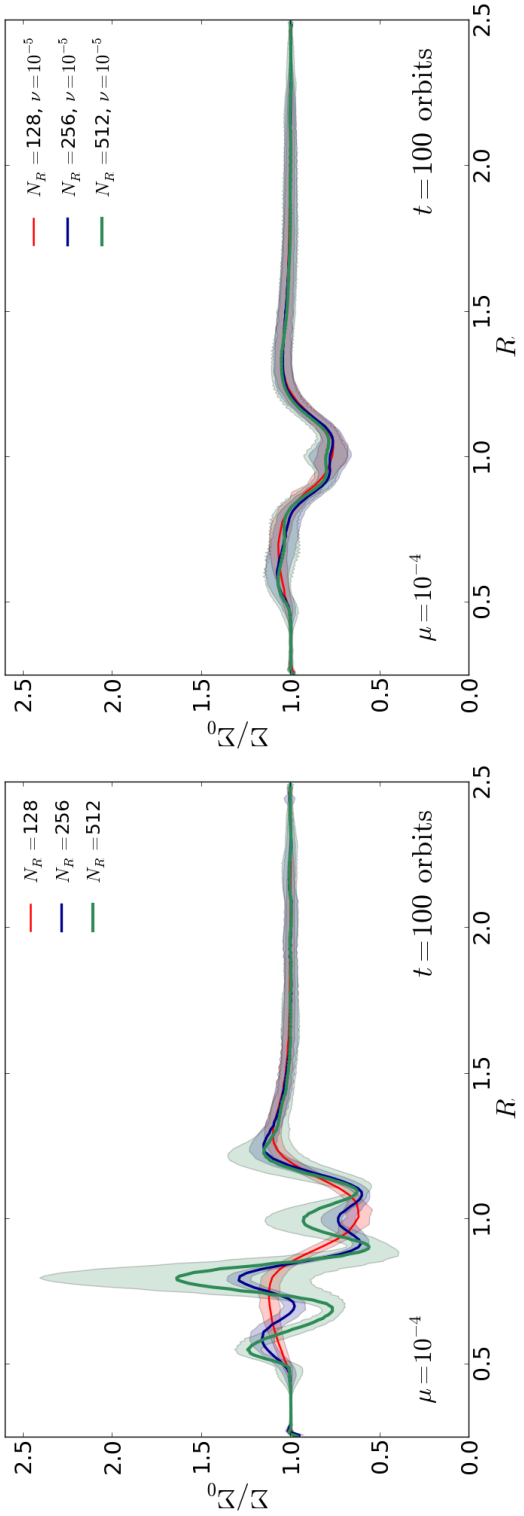


Figure 4.16.— Azimuthally-averaged gap profiles at $t = 100$ orbits with $\mu = 10^{-3}$. In addition to the azimuthal means (solid curves), 1-σ (one standard deviation) regions (shaded areas) are added to quantify deviations from axial symmetry. Top panel: profiles for inviscid simulations at three different resolutions, with $N_R = 128$ (red), 256 (blue) and 512 (green). Bottom panel: profiles for viscous simulations ($\nu = 10^{-5}$) at the same 3 resolutions.

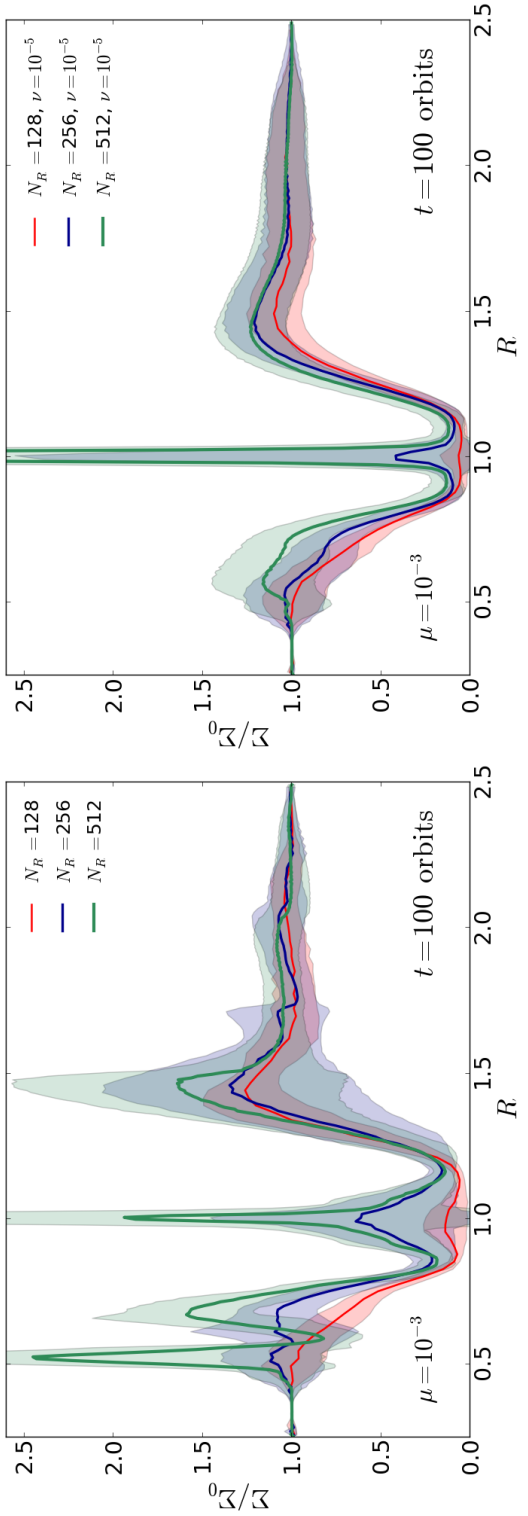


Figure 4.17.— Same as Figure 4.16 but for a mass ratio of $\mu = 10^{-3}$.

case. In addition, convergence shows a slower trend than in the viscous Neptune case. This is probably due to the fact that the edge instabilities are not entirely suppressed by a viscosity of $\nu = 10^{-5}$, and that the gap is deep enough to make the instability effective. By comparison, the very viscous runs ($\nu = 10^{-4}$) shown in Figure 4.15 (second row from the top) show an even greater degree of smoothness, and no evident dependence on resolution.

These results allow us to compare, at least qualitatively to the code comparison project of de Val-Borro et al. (2006), even though our setup is not identical (we have run globally isothermal simulations). Our results show broad consistency with the grid code output, and bear no resemblance to SPH results. This is a natural consequence of AREPO being a grid code, and that the quasi-Lagrangian nature of the moving-mesh scheme implies by no means behavior similar to SPH. One difference between our AREPO runs and the grid code simulations of de Val-Borro et al. (2006) is the the large peak in Σ at $R = 1$ in Figure 4.17 for both inviscid and viscous runs, which is due to the very high density of gas at the planet position. This effect appears to be enhanced with increasing viscosity. Since high-viscosity runs (e.g. second row from the top in Figure 4.15) suppress the formation of deep gaps, the gas supply onto the planet is not halted and thus accretion can proceed sustainedly. This accretion of gas translates to a high concentration of cells within the planet's Hill sphere (Figure 4.18).

Interestingly, this property of moving-mesh schemes opens new possibilities to overcome the strict resolution requirements imposed by the detailed hydrodynamics of gap clearing. For example, Figure 4.18 shows a high concentration of cells close to the planet, which allows for the study of circumplanetary disks within *global* circumstellar disk simulations, combining a coarse large scale disk, with a very high resolution Hill

Figure 4.18.— Density field and mesh structure of a viscous Jupiter run (simulation `JUP-VISC-B_512`) close to the planet. Top panel: the visualized region covers a square area of side length equal to 0.5, centered on the planet position at $t = 100$ orbits. Bottom panel: zoomed-in region of side length equal to 0.14, roughy twice the planet's Hill sphere. The concentration of cells toward the potential well minimum increases the resolution within the Hill sphere roughly by a factor of $\sim 20^2$-30^2 respect to the initial cell spacing.

sphere region, thus concentrating all the computational power in the regions of interest. Similarly, Figure 4.18 shows how the spiral wakes also concentrate a larger number of cells than the background flow. This property of moving-mesh methods can provide an alternative to the extremely high resolution studies that have been carried out by Dong et al. (2011b,a) and Duffell & MacFadyen (2012) concentrating the resolution elements specifically where the wake is.

Although all the simulation presented in this work make use of global time-steps, the possibility of studying circumplanetary disks in the future with AREPO will make it necessary to separate the domain time-steps as described by Springel (2010a), thus making calculations more efficient as well as avoiding the accumulation of truncation error by integrating at time-steps well below the Courant criterion. Indeed with global time steps, unless derefinement is allowed within the Hill sphere (for example, imposing a minimum cell volume), the small cells that can be seen at the planet location make the computation unjustifiably expensive for the resolution of the global disk. In the Jupiter-mass case, the Hill radius is $(\mu/3)^{1/3} \sim 0.07$, corresponding to 4, 8 and 16 cells across at $t = 0$ for the runs with $N_R = 128$, 256 and 512 respectively. However, after 100 orbits, the Hill sphere can have up to 200 cells across in the viscous run with $\nu = 10^{-5}$ and $N_R = 512$. If the entire simulation had been run with a globally uniform cell size like that obtained within the Hill sphere, the number of radial zones would have been $N_R = 6500$, a resolution comparable to the one used by Dong et al. (2011b,a) and Duffell & MacFadyen (2012).

4.3.3 Vortensity Field

Figure 4.11 shows the dramatic difference in absolute vorticity (i.e., the vorticity measured in the inertial frame) between a gapped and an non gapped disk. The local maxima in vorticity in the Jupiter-mass simulation are found at the edge of the gap, and are thus associated with peaks in the density and pressure profiles.

Pressure bumps are a known cause for a hydrodynamic instability in Keplerian disks known as the Rossby wave instability (RWI; Lovelace et al., 1999; Li et al., 2000, 2001; Tagger, 2001), an effect that eventually saturates into vortices. Thus, provided the pressure bump associated with the presence of a gap is large enough (Li et al., 2000), the presence of vortices at the edge of gaps is to be expected in low-viscosity planet-disk interaction simulations (see de Val-Borro et al., 2007; Lyra et al., 2009). Thus, the steeper the variation in density at the edge of the gap (see Figure 4.17, top panel) the easier the formation of vortices.

Lovelace et al. (1999) and Li et al. (2000) found that a Keplerian disk is unstable to azimuthal perturbations when the following quantity reaches a local extremum

$$\mathcal{L}(R) \equiv \mathcal{F}(R)(P\Sigma^{-\gamma})^{2/\gamma} \quad .$$

In a barotropic disk, $\mathcal{F}(R) = \Sigma\Omega\kappa^{-2}$, where Ω is the orbital angular velocity and $\kappa = [R(d\Omega^2/dR) + 4\Omega^2]^{1/2}$ is the epicyclic frequency, which can be related to the z-component of the vorticity for *axisymmetric flow* by $\omega_z = (1/R)\partial(R^2\Omega)/\partial R = \kappa^2/(2\Omega)$. Therefore, for the globally isothermal case (i.e., $\gamma = 1$)

$$\mathcal{L}(R) \propto \frac{\Sigma}{2\omega_z} = \frac{1}{2\zeta} \tag{4.10}$$

where $\zeta \equiv \omega_z/\Sigma$ is the vortensity or "potential vorticity" of the disk. Equation 4.10

implies that local extrema in the radial vortensity profile can trigger the RWI. A saturated RWI should be very efficient at destroying the extremum in vortensity (e.g. Meheut et al., 2010), however, the presence of a massive planet can replenish the density bump observed in Figure 4.17, enabling a sustained production of vortices (Lyra et al., 2009).

Creation of vortices outside the density bump by virtue of a saturated RWI is not the only mechanism to generate non-axisymmetric vortical structures. Let us recall that barotropic Rossby waves conserve vortensity along Lagrangian trajectories of fluid. Indeed, one can combine Equations 4.1a and 4.1b into a single equation for vortensity

$$\frac{\partial}{\partial t}\left(\frac{\omega_z}{\Sigma}\right) + \mathbf{v}\cdot\nabla\left(\frac{\omega_z}{\Sigma}\right) = \frac{\nabla\Sigma\times\nabla P}{\Sigma^3} \quad , \tag{4.11}$$

which implies that $D(\zeta)/Dt = 0$ for barotropic disks. However, a way to "produce" vorticity directly is by means of oblique shocks, even in barotropic fluids (Truesdell, 1952; Kevlahan, 1997). This mechanism is suggested by Lin & Papaloizou (2010) to produce sharp vortensity rings caused by the spiral waves launched by the planet. According to Lin & Papaloizou (2010), these vortensity rings are dynamically unstable, and saturate into vortices *within* the coorbital region, which can scatter off the planet, causing rapid migration events in what is one version of the so-called runaway, or Type III migration (see also Masset & Papaloizou, 2003; Artymowicz, 2004; Papaloizou, 2005)

Figure 4.19 shows the vortensity field of the fiducial ($N_R = 128$) inviscid runs in the frame *rotating* with the planet divided by the vortensity field at $t = 0$. The left panel shows the vortensity for the Neptune-mass run while the right panel shows the vortensity for the Jupiter-mass runs. Since our simulations are run using an inertial reference frame, we obtain the *absolute* vorticity $\nabla\times\mathbf{v}$ directly from the interpolated

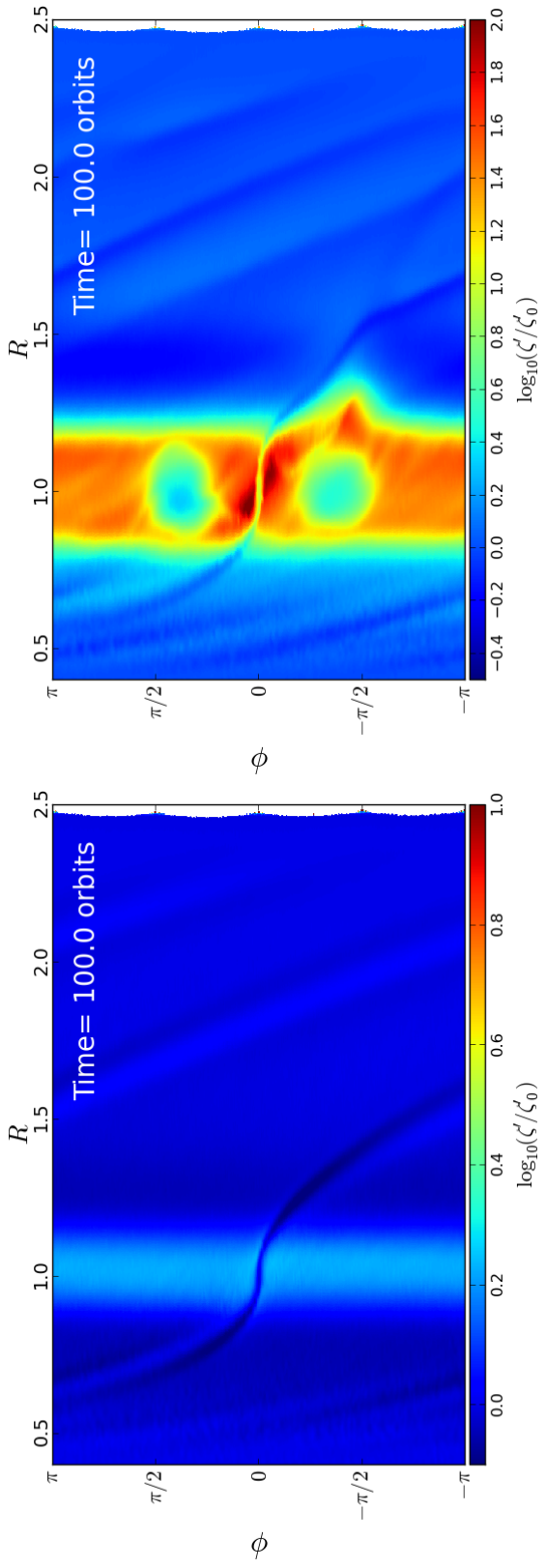


Figure 4.19.— Vortensity field in the rotating frame of the planet in polar coordinates normalized to the initial vortensity field ζ_0. Left panel: fiducial inviscid Neptune simulation ($N_R = 128$). Right panel: fiducial inviscid Jupiter simulation ($N_R = 128$). The color table and color range are chosen to closely match the vortensity fields plotted in de Val-Borro et al. (2006).

velocity field. In order to obtain the *relative* vorticity (i.e., the vorticity as seen in the frame rotating with the planet) we simply subtract $2\Omega_p$, i.e., the vorticity associated with the background flow:

$$\zeta' = \frac{\nabla \times \mathbf{v}'}{\Sigma} = \frac{\nabla \times \mathbf{v} - 2\Omega_p \hat{\mathbf{k}}}{\Sigma} \quad . \tag{4.12}$$

Since we are now measuring derivatives of the primitive quantities of the flow, it is expected that the vorticity field will converge more slowly than the density field (AREPO uses a linear reconstruction strategy).

Figure 4.19 shows some differences with respect to the sharper vortensity results of other cylindrical grid codes shown by de Val-Borro et al. (2006). Indeed, the vortensity peaks near the edges of the partial gap present in the Neptune simulations are shallower in our example. In addition, we see less structure within the gap. The Jupiter run, on the other hand, does show a sharp transition in vortensity across the edge of the gap, and succeeds in capturing the vortensity "islands" at the L_4/L_5 Lagrangian points, which is not achieved by all codes. Also, the vortensity field is devoid of reflections from the boundaries, which shows that our absorbing boundary condition is effective at eliminating such artifacts. Some differences include the saturation of some of the contours in de Val-Borro et al. (2006) at the lower end of the color range ($\log_{10}(\zeta'/\zeta'_0) = -0.5$) where vortices form outside the gap, an effect that is more subtle in the right panel of Figure 4.19. By contrast, the vortensity contours in our run saturate at the upper end of the color range ($\log_{10}(\zeta'/\zeta'_0) = -0.5$) close to the planet, an effect that is not seen in the results of de Val-Borro et al. (2006), and which we attribute to the increased resolution close to the planet that the moving mesh allows.

To test how converged the results of Figure 4.19 are, we calculate the normalized

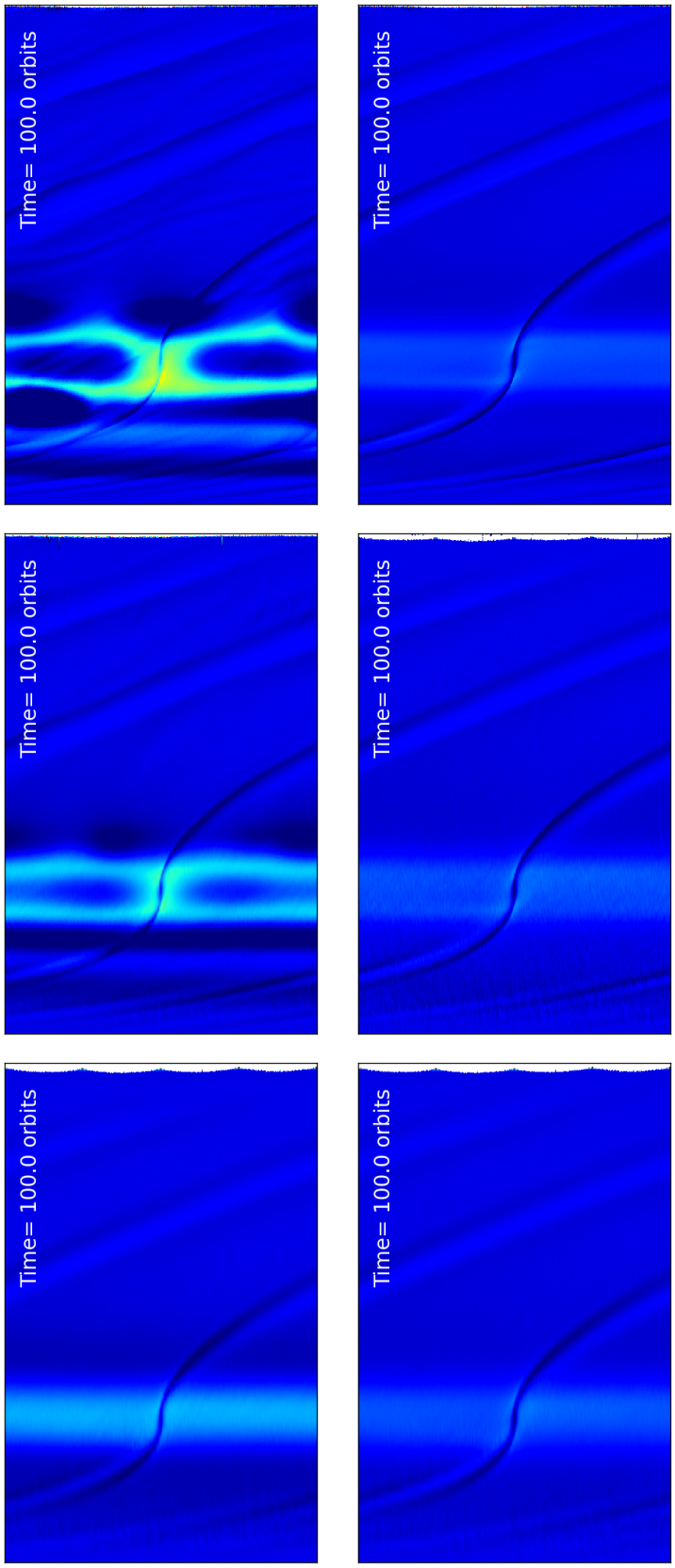


Figure 4.20.— Deviations in the vortensity field (time=100 orbits) for the Neptune-mass simulations at three different resolutions including the inviscid and viscous cases. Top panels: inviscid Neptune runs with resolutions (left to right) N_R =128, 256 and 512. Bottom panels: viscous ($\nu = 10^{-5}$) Neptune runs with resolutions (left to right) N_R =128, 256 and 512. Color scale and coordinate axes are identical to that of the left panel of Figure 4.19, which is shown again in the top-left panel of this figure.

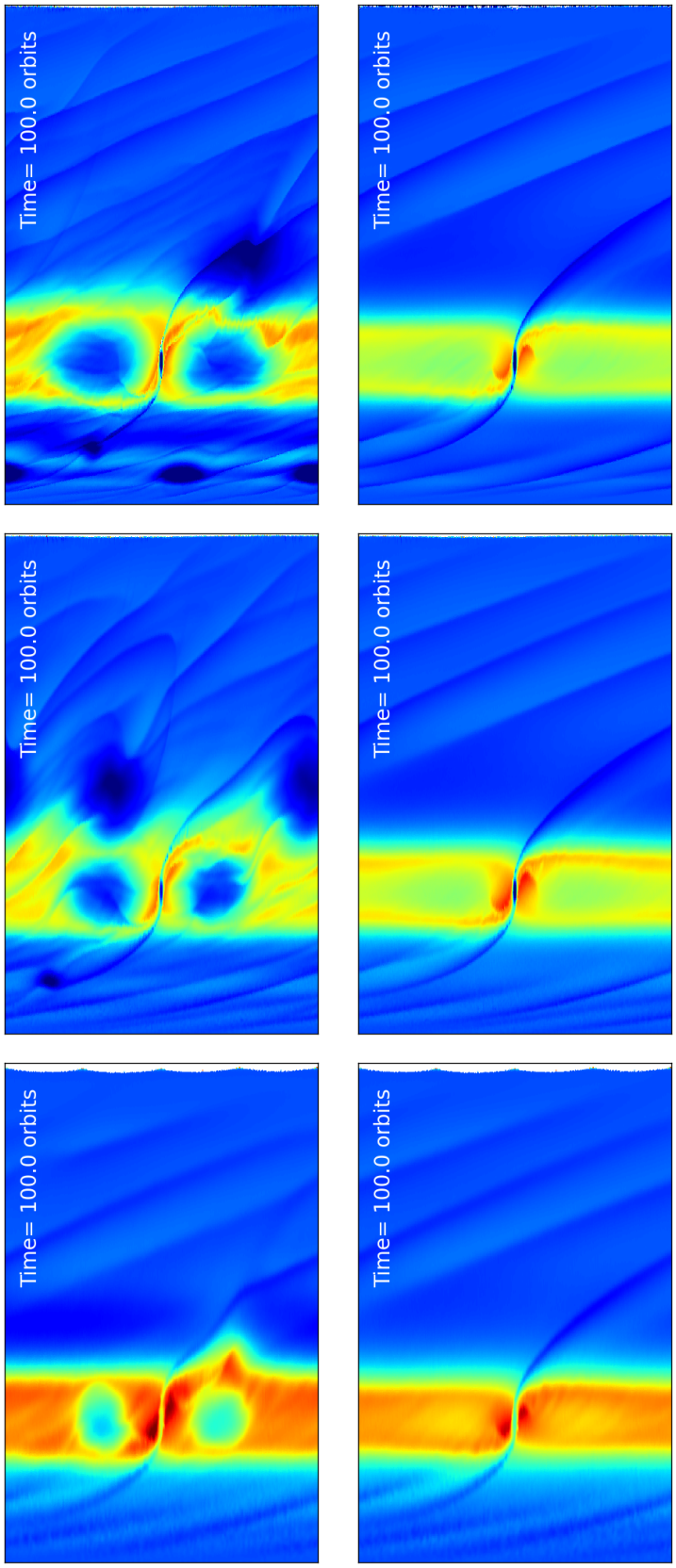


Figure 4.21.— Same as Figure 4.20, but for the Jupiter-mass runs. Top panels: inviscid Jupiter runs with resolutions (left to right) N_R =128, 256 and 512. Bottom panels: viscous ($\nu = 10^{-5}$) Jupiter runs with resolutions (left to right) N_R =128, 256 and 512. Color scale and coordinate axes are identical to that of the right panel of Figure 4.19, which is shown again in the top-left panel of this figure.

vortensity field ζ'/ζ'_0 for three different resolutions in the inviscid Neptune, the viscous Neptune, the inviscid Jupiter and the viscous Jupiter cases. The normalized vortensity fields are shown in Figure 4.20 for Neptune and Figure 4.21 for Jupiter. As discussed in Section 4.3.2, the inviscid runs should be expected to converge slowly, since higher resolution runs will be effectively less diffusive, and the appearance of vortices should increase with increasing resolution, as a consequence of shaper gaps. Indeed, both inviscid examples (top rows in Figures 4.20 and 4.21) show an increase in the number of vortices every time resolution is augmented. The inviscid Neptune runs at $N_R = 256$ and $N_R = 512$ show the vortensity rings described by Lin & Papaloizou (2010). The different vortensity fields for the viscous runs are nearly indistishable from each other, confirming that a viscosity of $\nu = 10^{-5}$ is enough to suppress the shock-induced generation of vortensity.

We do expect the viscous runs to show some degree of convergence. Interestingly, there is consistency in between $N_R = 256$ and $N_R = 512$, for both the Neptune and Jupiter cases, but the $N_R = 128$ clearly stands out as unconverged. This confirms our previous observation that 128 radial zones might not be enough to capture the global flow properly, especially if vortices are expected to develop. As mentioned before, the derivatives of the primitive variables are at best first-order accurate when linear reconstruction is implemented.

The vortensity features in inviscid runs with $N_r = 256$ (for both Neptune and Jupiter examples) are significantly sharper and richer than the those of de Val-Borro et al. (2006). Therefore, we believe that to achieve results comparable to the most robust results of de Val-Borro et al. (2006), we need a number of radial zones lower than 256 but higher than 128. We also point out that some of the differences between our AREPO

runs and others might be caused by the varying azimuthal resolution with radius; or simply the fact the the effective azimuthal resolution at $R = R_p$ is ~ 360, still below the $N_\phi = 384$ used by the simulations of de Val-Borro et al. (2006); or by other numerical effects associated with the orbital evolution of the cells using a KDK integrator, which might add noise to the velocity field.

4.3.4 Total Torque Evolution

Finally, we consider the measurement of the tidal torques. Ultimately, a successful planet-migration simulation will be determined by the accurate calculation of the tidal torques since these determine the rate of change of the planet's angular momentum. The tidal torque exerted by the disk onto the planet is

$$\begin{aligned} \mathbf{T} =& \mathbf{r}_p \times \int_{\text{disk}} \mathbf{f}\, \Sigma(\mathrm{r})\, dA \\ =& GM_p\, \mathbf{r}_p \times \left[\int \Sigma(\mathbf{r}) g_1 \left(|\mathbf{r}'| \right) \mathbf{r}' dA \right] \quad , \end{aligned} \tag{4.13}$$

where $\mathbf{r}' = \mathbf{r} - \mathbf{r}_p$ and $g_1(y)$ is the spline-softened gravitational force function (Springel et al., 2001), which evaluates to the exact Keplerian value of $1/y^3$ for $y \geq h = 2.8\epsilon$, with ϵ being the tradtional Plummer softening parameter. We discretize Equation 4.13 as

$$\mathbf{T} = GM_p\, \mathbf{r}_p \times \left[\sum_{i=0}^{N} M_i g_1 \left(|\mathbf{r}'| \right) \mathbf{r}' \right] \quad , \tag{4.14}$$

summing over all gas cells of masses M_i, and using $\mathbf{r}' = \mathbf{r}_i - \mathbf{r}_p$, where $\mathbf{r_i}$ is the cell centroid. The cells within the planet's Hill sphere –i.e., $|\mathbf{r}'| \lesssim (\mu/3)^{1/3}$– are excluded from the sum in Equation 4.14.

Figure 4.22 shows the torque evolution as computed from Equation 4.14 at each snapshot. Separating between inner and outer torques (i.e. summing over cells inside

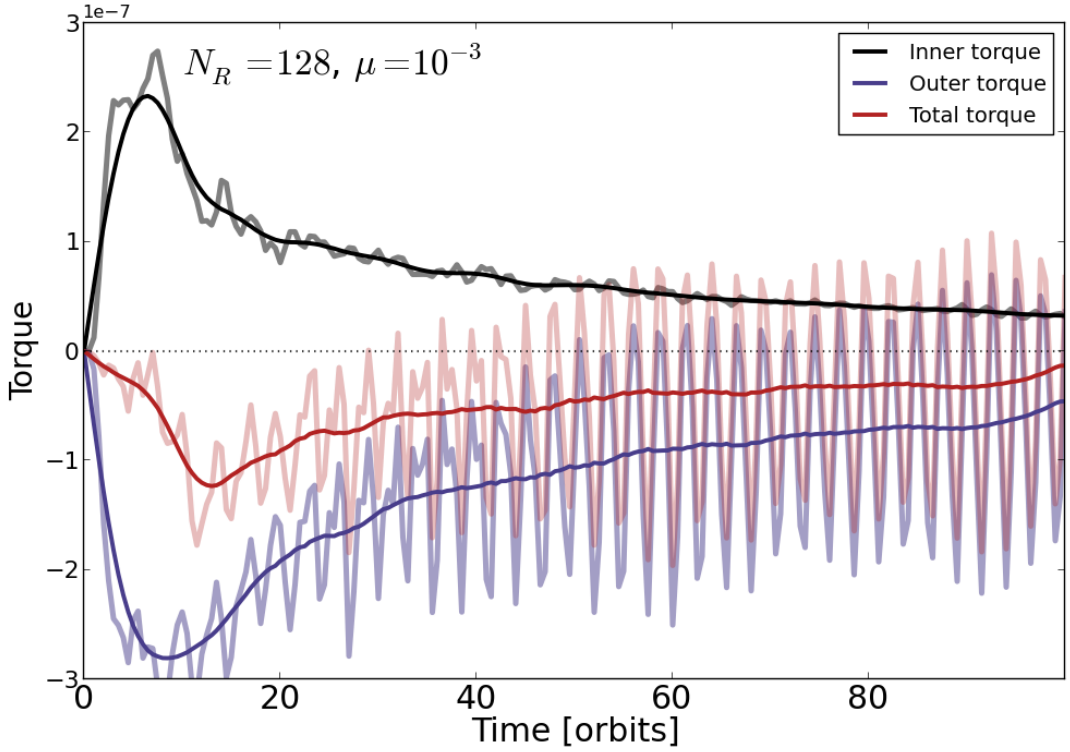


Figure 4.22.— Evolution of the tidal torque for the fiducial inviscid Jupiter simulation (run `JUP_128`).The figure includes the torques according to Equation 4.14 as a function of time at a sampling rate of 1 orbit (dark lines) as well as a smoothed evolution with a smoothing interval of 10 orbits using a Hanning window function (lighter curves). Black solid lines represent the torque evolution from gas cells at within $R = 1$, blue lines represent the torque evolution from cell outside $R = 1$, and the red curves represent to total torque. The fluctuations in the density field (Section 4.3.1) are large enough to change the sign of the outer torque and, therefore, the direction of migration. These oscillations are removed by averaging over a 10-orbit period.

and outside $R = 1$), it can be seen that the smoothed torque (lighter lines in the figure) is slightly negative, thus enabling an inward migration of the planet. The raw torque time series, however, tells a different story. The small scale asymmetric features observed in the density field (Section 4.3.1) are large enough that they can alter significantly the torque onto the planet, even to the point of changing its sign on timescales comparable with the planet's orbital period. These fluctuations, identified as vortices in our simulations, are a phenomenon observed previously by Koller et al. (2003), Li et al. (2005) and de Val-Borro et al. (2007). In some cases, the planet-disk torque can be dominated by planet-vortex scattering (Lin & Papaloizou, 2010) triggering a runaway migration sometimes referred to as type III migration (see also Masset & Papaloizou, 2003; Artymowicz & Lubow, 1994; Papaloizou, 2005). Since this type of migration depends sensitively on the vortensity field within the coorbital region, the exclusion of the Hill sphere from the torque calculation might produce drastically different results than what would be obtained in a fully consistent, self-gravitating planet-disk interaction (e.g. Crida et al., 2009).

Figure 4.23 shows that the tidal torque acting on a Neptune-mass planet is of comparable magnitude to the fluctuations. These fluctuations do not show the periodicity observed in the Jupiter counterpart and they are potentially due to numerical noise. Since the perturbation on the background density field is proportional to the planet mass, the torque resulting from this non-axisymmetric density field is fractionally noisier than that caused by a planet mass ten times larger, as is in the Jupiter case.

In order to assess the reliability of the torque calculation, we measure $\mathbf{T}$ at three different resolutions ($N_R = 128$, 256 and 512; see Table 6.1). Since we do not expect the inviscid runs to converge given the high sensitivity of this problem to diffusion

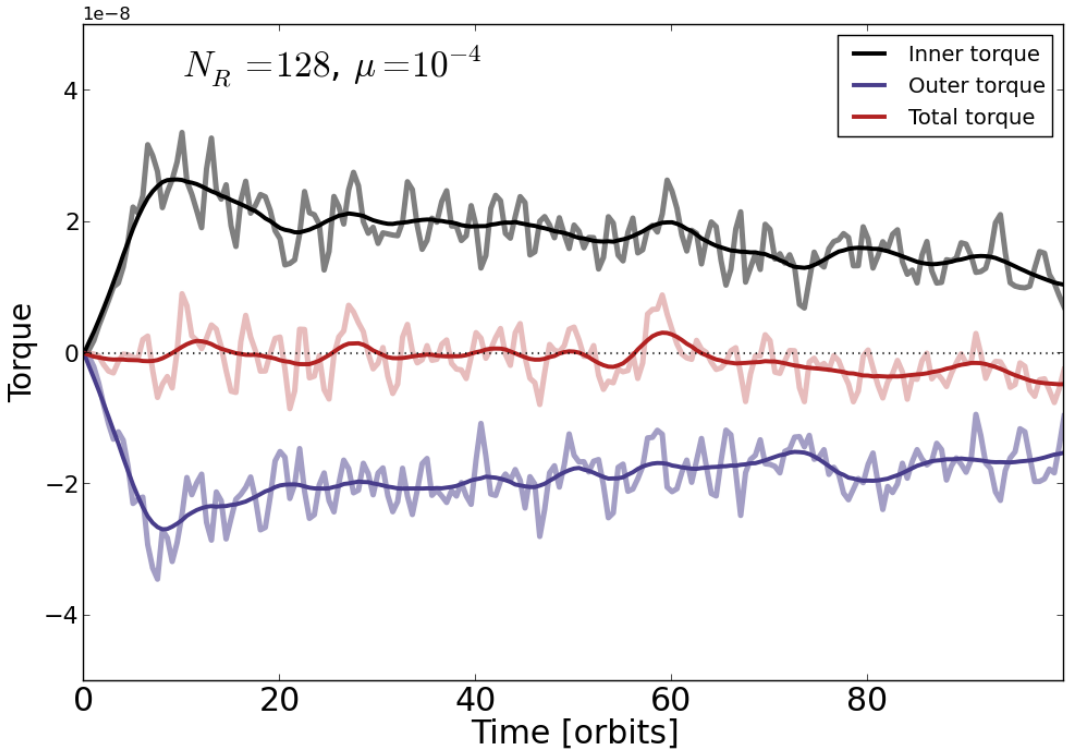


Figure 4.23.— Same as Figure 4.22 but for the fiducial inviscid Neptune-mass simulation (run `NEP_128`). In this case, the inner and outer contributions to the total torque are more symmetric than in the Jupiter-mass case. In addition, the fluctuations in the torque calculation are of the same order as the difference between the inner and outer torques, and thus an accurate total torque calculation is difficult at this resolution.

(whether it is numerical or physical), we explore the convergence of the torque evolution for the viscous runs with $\nu = 10^{-5}$. Figure 4.24 shows the torque evolution for the viscous Jupiter-mass runs at the three different resolutions listed in Table 6.1. Although the qualitative torque evolution is very consistent across the different resolutions, the transient period ($t \lesssim 50$ orbits) shows a non-converging behavior. Note that the location and shape of this variability, which does not vanish after applying a time-series smoothing, is not entirely inconsistent across the different panels. However, the amplitude of these transients grows with resolution, indicating that a viscosity of $\nu = 10^{-5}$ is not high enough to suppress vorticity generation near the coorbital region that can generate chaotic variations in the net torque. Note that the net torque at $t = 100$ orbits is very consistent across all resolutions, showing a robust convergence of the planet's migration rate. In the near-stationary regime ($t \gtrsim 50$ orbits), the individual components of the torque (the inner and outer contributions) are independently consistent with each other for the $N_R = 128$ and $N_R = 256$ runs. However, and despite a consistent net torque, the individual components show a discrepancy at $N_R = 512$. Higher resolution runs are required to settle whether these simulations are converged on all accounts.

Figure 4.25 shows a more convincing convergence than Figure 4.24. As mentioned above, the smaller perturbation exerted on the disk by a Neptune-mass planet implies that the torque magnitude can be more easily "buried" by the noise than in the Jupiter-mass case. Conversely, the qualitative quasi-stationarity of the Neptune case (e.g. Figure 4.9) provides more reassurance that convergence can be reached. Indeed, although a resolution of $N_R = 512$ is needed to beat down the fluctuations, convergence can be observed in Figure 4.25, showing that a slight asymmetry between the inner and outer torques will cause an inward migration of the planet.

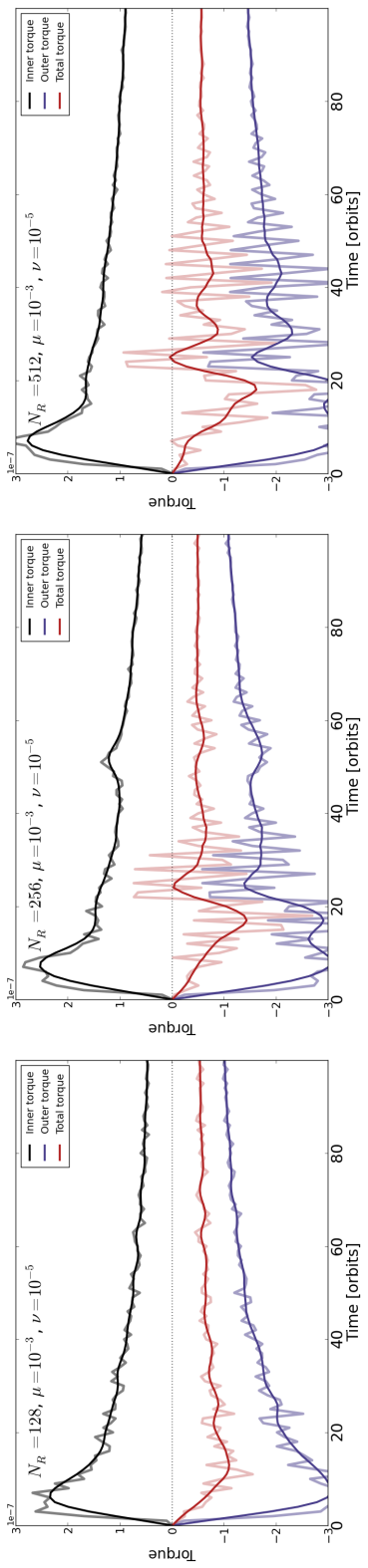


Figure 4.24.— Tidal torque evolution for a Jupiter-mass simulation with viscosity coefficient $\nu = 10^{-5}$ at three different resolutions: $N_R = 128$(top panel), 256 (mid panel) and 512 (bottom panel), corresponding to runs `JUP-VISC-B_128`, `JUP-VISC-B_256` and `JUP-VISC-B_512` respectively. The total torque at $t = 100$ orbits is consistent across all resolutions, although the magnitude of the inner and outer torques of the highest resolution run differs from the rest. The transient stage of the torque calculation (up to ~ 50 orbits) shows no hints of convergence.

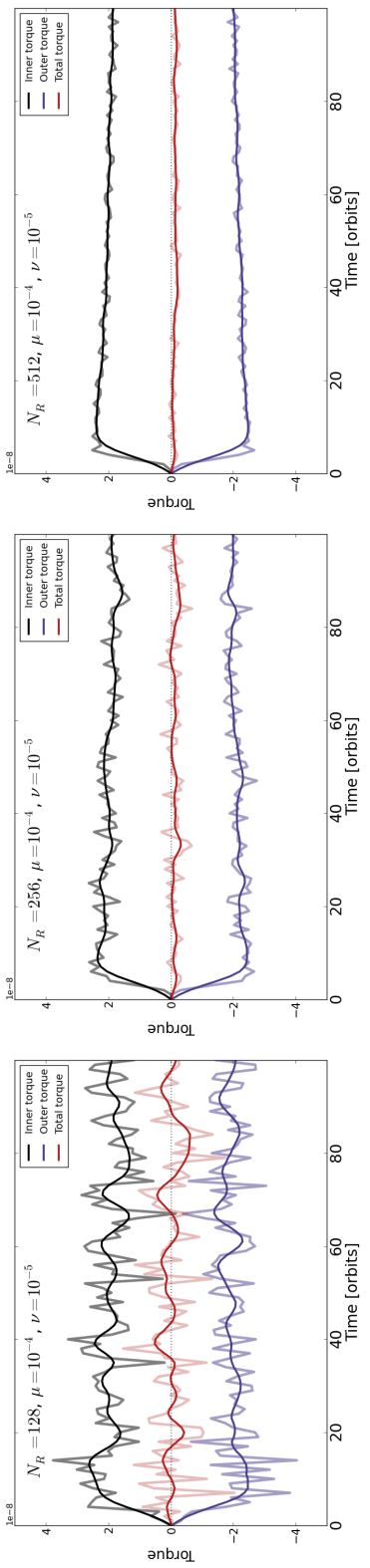


Figure 4.25.— Same as Figure 4.24 but for a Neptune-mass planet (runs `NEP-VISC-B_128`, `NEP-VISC-B_256` and `JUP-VISC-B_512`, from top to bottom). Convergence is achieved, albeit slowly with resolution. There is a tradeoff in comparison to the Jupiter-mass case: since the density fluctuations due to the planet wake are weaker, the smaller the planet mass, the higher the resolution is needed to recover the theoretically expected negative net torque; on the upside, the smoother variation of this configuration (there is no gap opening) guarantees that a consistent solution can be reached with high enough resolution, as opposed to the transient-plagued torque evolution of Figure 4.24.

4.4 Discussion and Final Remarks

Applying AREPO to an idealized problem of planet-disk interaction provides a very important opportunity to benchmark the moving-mesh method in a setup which we anticipate is challenging for this type of code.

We have observed results that are very consistent with simulation results in the literature, contradicting recent claims that a code like AREPO has intrinsic limitations for problems requiring a high-degree of symmetry, especially with a supersonic shearing flow (Duffell & MacFadyen, 2012). We do acknowledge that when the flow is very close to axisymmetric, AREPO can be at best competitive with static grid methods, owing to the additional grid noise provided by the moving mesh. Such a code will naturally be less efficient, given the additional computational expense of retessellating at every time-step.

We have observed that even higher resolution will be needed in future work to assess the convergence of the torque calculation for Neptune-mass planets, let alone for Earth-mass planets. We attribute the noise component of the net torque evolution observed in the Neptune case to fluctuations in mass and position of the cells. Future work should explore new ways to drift the under a point-mass potential. An interesting possibility is to drift the mesh generating points using a higher order integrator than a leapfrog, but of similar conservative properties, such as the Wisdom-Holman mapping used for planetary systems (Wisdom & Holman, 1991). In addition, new algorithms to regularize the mesh should be explored. At this point, mesh regularization has –unsurprisingly– degrees of freedom that do not necessarily depend on the flow evolution. Moving toward parameter-free regularization techniques is highly desirable.

A potential source of error in the torque calculation can originate from treating the cells as particles. Let us recall that the mesh itself should be interpreted physically with caution, since it is not a fluid quantity per se. However, when computing gravity in an N-body fashion (as we have done to compute the tidal torque in Equation 4.13) we have done precisely that. In addition, gravity is included as a source term, thus it unavoidably increases the truncation error with respect to a flux-based gravity treatment.

In general, we identify three features intrinsic to this type of code that can present a limitation to its efficient performance in the problem of planet-disk interaction. The first one is the inclusion of gravitational acceleration in the form of a source term that is calculated in an N-body fashion, i.e., interpreting each cell as a particle of (nearly) constant mass. The second one is the implementation of the moving mesh itself. Since the mesh is evolved and not fixed in time, it introduces both grid noise and truncation errors. This effect, coupled to the first one, can lead to inaccuracies in the torque calculation. Finally, the third property, and the most obvious one, is the non-conservation of angular momentum in an unstructured mesh. This last feature is of little concern on its own, since convergence has been reached to achieve its theoretical value for rotationally supported flows (Springel, 2011).

We believe that the features listed above are common to all tessellation-based approaches. Alternatives might include the flux-based gravity approach attempted by Springel (2010a) for earlier versions of the AREPO code, which is similar to the one presented by Jiang et al. (2013) or the different mesh-drifting scheme of Duffell & MacFadyen (2011) for the two-dimensional TESS code. There are, however, algorithmic differences between different Voronoi-tessellation codes that can change entirely their performance despite their similarities in moving-mesh approaches. One difference

between TESS and AREPO is, for example, that TESS does not use the MUSCL-Hancock method to achieve second-order accuracy in space and time, but the method of lines, which tightly couples the evolution of the mesh-generating points with the evolution of the hydrodynamical state vector for the half-time-step predictor stage. This difference, in turn, forces TESS to re-tessellate at the middle of the time-step, which is not a requirement for AREPO. However, the biggest difference between these two approaches is the critical interplay between the moving mesh and the Riemann solver, and how these two elements of the numerical method define the numerical diffusion of the code.

It is known that the so-called "false diffusion" (or "numerical viscosity" or "advection error") on the grid/mesh arise four different ways. The most obvious one is the grid spacing. The least obvious one is grid noise. The other two are the fluid velocity *with respect to the grid*, and the angle between the flow direction and the coordinate axis used to discretize the domain (de Vahl Davis & Mallinson, 1972; Patankar, 1980), i.e. the alignment of the grid with the flow. Although these last two features are interrelated, they are not quite equivalent. Below, we address them separately.

Alignment of the mesh in one direction might come at the expense of misalignment in another direction. As discussed briefly by Duffell & MacFadyen (2012), a supersonic shearing flow might not be ideal for a Voronoi mesh since contiguous cells at different speeds will share a rapidly rotating interface that will not be parallel to the direction of the flow at that point, thus eliminating the benefit that meshes that are aligned with the flow have in reducing numerical diffusivity. Indeed, the exchange of mass between cells moving along the same axis in the direction of the bulk motion will be minimal. However, two cells moving in the same direction but along two different axis at different speed will experience some mutual shear. Since the Voronoi tessellation always puts the

normal to the interface parallel to the line connecting the two cell centers, these two cells will exchange mass since the bulk flow will not be perpendicular to the face normal, as it would have been if the mesh had remained cartesian. Indeed, this amounts to additional diffusion. Although a symmemtric enough mesh will balance out the exchange of mass, truncation error will prevent the Godunov fluxes from remaining close to the roundoff noise.

This misalignment is intrinsic to using a Voronoi tessellation, and is thus unavoidable when using a freely moving mesh. As a result, Duffell & MacFadyen (2012) decided to revert to a structured grid approach in which, although cells are indeed moving, their motion is restricted to be along the azimuthal direction, in a technique highly reminiscent of the original FARGO scheme. However, there is one subtlety to this problem. While it is true that this shear-induced diffusion cannot be minimized to the levels that would be reached if the projected flux over a given face were identically zero (which is the case of the interface between two contiguous radial zones in a polar grid), it can be minimized. It is possible to reduce the advection error greatly if the motion of the rotated face is subtracted. If the mesh were not moving, a static Voronoi mesh would combine the worst of both worlds: a large bulk motion with respect to the cell interfaces, and a randomized misalignment that would make the code's performance poorer than that of a fixed cartesian grid. However, if the bulk motion is taken into account, the problem is greatly alleviated. This requires a boosted Riemann solver, which is not included by Duffell & MacFadyen (2011) in the original version of TESS.

Now we turn to the second source of advection error: the bulk motion of the flow with respect to the grid. In practice, AREPO takes into account the motion of the mesh by adding the respective Godunov fluxes at each interface assuming the mesh is *at rest* at

any given time, following with the subtraction of an advection flux that compensates the fact that the face is moving. A crucial detail in the AREPO scheme is how the Godunov fluxes are obtained in the first place. In AREPO, the Riemann problem is solved in a *boosted* frame (i.e., the flow velocity to the left and right of the interface is always small), and the resulting primitive variables are boosted back into the lab frame. Only then are the Godunov fluxes are calculated and the quantities are updated. On the other hand, TESS solves the Riemann problem directly in the lab frame, which means that the velocity flow to the left and right of the interface can be very large. Even if the posterior subtraction of the advection flux reduces the mass exchange between two contiguous cell, the truncation error has already done its damage and the scheme will be diffusive. Note that Duffell & MacFadyen (2011) acknowledge this problem, but dismiss it as their code is simply "not Galilean invariant", a feature that they estimate desirable but not necessary. However, the reason for the violation of the so-called Galilean invariance is simply that a lab-frame Riemann solver is very diffusive, even if the mesh is allowed to move in order to better capture contact discontinuities. Duffell & MacFadyen (2011) attribute the lack of Galilean invariance of their code to their choice of an approximate Riemann solver. However, the use of an approximate Riemann solver has nothing to do with "Galilean invariance", as demonstrated by Pakmor et al. (2011), as long as the Riemann problem is properly boosted to the face rest frame.

Let us recall that what Springel (2010a) (controversially) refers to as "Galilean invariance" is really a reduction of the advection error thanks to the Galilean boost of the Riemann problem plus an geometrically flexible moving mesh. As discussed by Springel (2010a), even a boosted Riemann solver will stop producing Galiean-invariant simulation output for *extremely* high-Mach number flows if the boosting operation itself

(which changes the primitive variables from the lab frame to the face frame and back into the lab frame) introduces significant errors to the evolved quantities, i.e., when the *roundoff* error is as important as the truncation error that the boosting procedure aims to minimize. This is a *very* important point, and an essential element to the moving-mesh approach: if the Riemann solver is not boosted to the face frame, the fact that the mesh is moving only reduces the advection error by adapting the alignment of the face (therefore improving the capture of contact discontinuities with respect to a fixed-grid approach) and thus only minimizing one of the two sources of advection error we have discussed here. However, high-Mach number flow (like an accretion disk) solved without boosting the Riemann solver will provide marginal benefits at a much greater computational cost. Although no TESS results have been published for accretion disk simulations, the perception that it does not perform adequately for accretion disks simulations is related to the way the Riemann solver has been implemented, and not to the geometry of the mesh.

Unfortunately, we have not been able to produce a successful simulation with a lab-frame Riemann solver on a moving mesh. As described in detail by Pakmor et al. (2011), such a scheme should be unstable since it can easily violate upwinding of the fluid quantities. Since Duffell & MacFadyen (2011) did not report any errors of that sort, we assume that their implementation of the method of lines with a mid-time-step tessellation make their scheme more apt to he use of a lab-frame Riemann solver.

4.5 Summary

We have presented results for low-resolution simulations of planet disk interaction using the moving-mesh code AREPO for two different planet-to-star mass ratio exploring the dependence of the result on resolution and viscosity.

1. We have shown that AREPO can work adequately with problems of high degree of symmetry like that of planet disk interactions, even though this is not what a code like this is intended to excel at. Although concerns about the numerical noise associated with faces being misaligned with the flow is well-founded, they do not affect the overall performance of the code. We argue that lab-frame Riemann solvers are inadequate for moving-mesh simulations, and that the Riemann solver (exact or approximate) should always be boosted to the frame of the face.

2. Among the different sources of noise, we conjecture that grid noise is the main concern in moving mesh simulations due to its sensitivity to the development of instabilities.

3. We have found that proper convergence of the simulations is a function of planet mass. This is not surprising since the perturbations exerted on the disk are proportional to the planet mass (in the linar regime), and thus it is easier to overcome numerical fluctuations with larger planetary masses

4. The quasi-Lagrangian nature of a code like AREPO opens new possibilities to the high-resolution study of planet-disk interaction. The possibility of a very flexible increment of resolution around areas of interest presents an efficient alternative to

uniformly increase the resolution globally, with the computational costs that this entails.

5. Although the merit and success of FARGO and FARGO-like codes is undisputable for this kind of problems, we believe there is room for moving-mesh codes, especially for tackling the adaptive mesh refinement difficulties that arise in cylindrical coordinates, but that are straightforward for AREPO

5

Circumstellar Disk Models in Isolation with Self-gravity

5.1 Introduction

Although the formation of a flattened, rotating structure around a young stellar object is a natural consequence of angular momentum conservation during gravitational collapse of a dense core, direct *resolved* imaging of such objects was not possible until 1990s. (O'dell et al., 1993; McCaughrean & O'dell, 1996; Mundy et al., 1996; Wilner et al., 1996). The evident challenges in spatially resolving these objects has begun to be overcome in the last decade, particularly thanks to (sub-)millimeter interferometry, which has allowed astronomers to map the fainter sources and to resolve disk sizes and inner cavities (e.g. Hughes et al., 2008, 2009) .

Up until recently, disk images have not had sufficient resolution to show fine

structure (at sub-AU scales). As a consequence, the observed surface densities have been successfully modeled in simple parametric form. The level of complexity of these analytic models has varied little since the early theory of accretion disks (e.g. Shakura & Sunyaev, 1973; Pringle, 1981), nevertheless providing enough physical insight into the structure of protoplanetary disk, such as measurements of density profiles and temperature profiles (e.g. Andrews et al., 2009, 2010b) and the disk-based dynamical determination of stellar masses (Rosenfeld et al., 2012). However, recent observations in mid-IR scattered light (Muto et al., 2012; Grady et al., 2013) and CO/HCO^+ sub-millimeter emission (Casassus et al., 2013) indicate that the finer structure of circumstellar disks can be quite rich and deviate significantly from axisymmetry. The limitation of parametric models can be overcome by direct numerical simulation, which enables us to capture the gas dynamical evolution of these systems self-consistently and study complex configurations in detail.

Besides the deviations from axisymmetry, there are other reasons why hydrodynamical modeling of these systems is necessary. One example is that of planet-disk interaction (e.g. de Val-Borro et al., 2006), especially in the presence of more than one planet. One of the most striking features of the *Kepler* catalogs (e.g. Batalha et al., 2013) is the widespread presence of multi-planet systems. Multi-planet systems add a significant degree of complexity to some of the physical processes believed to be relevant for the formation and survival of planets, namely planet migration and planet growth through the core and gas accretion phases. Planet formation is not only plagued with the effects planet multiplicity, but it is also likely to be affected to some extent by the environmental *stellar* multiplicity. Multiplicity rates of pre-main sequence stars are known to be higher than those of main sequence stars (Mathieu, 1994; Kraus et al., 2011), and this must have some influence on the primordial circumstellar disks and

the subsequent dynamical evolution of planets. Recent observations of young multiples in Taurus with both the *Spitzer* Infrared Spectrograph (IRS) (e.g. Kraus et al., 2012) and the Submillimeter Array (SMA) (e.g. Andrews et al., 2010a; Harris et al., 2012) reveal increasingly diverse multiples and binaries bearing circumstellar and circumbinary disks, it becomes evident that the idealized system consisting of one star and one disk might not be representative of the Galactic-wide planet formation process. Despite these increasingly complex configurations, hydrodynamical simulations of planet-forming systems often focus on models of gas disks orbiting one isolated, stationary star.

Even in light of their great flexibility and ability to capture time-evolving gas dynamics consistently, numerical simulations are also subject to a series of limitations. Circumstellar disks within star-forming environments (including, among other conditions, stellar multiplicity and stellar bulk motion) can be extremely challenging to some of the known numerical techniques. One of such challenges has to do with the way the geometry of the system determines which set of coordinates is most appropriate for the formulation of the hydrodynamic equations and their subsequent discretization. It is well known that the performance of Eulerian hydrodynamical codes depend significantly of the geometry of the mesh aligning with the direction of bulk flow [1] . Truncation error analysis allows for the derivation of "model" or "modified" equations (see LeVeque, 2002, §8 and references therein) of which the discrete version of the equation of motion are exact solutions, showing that space and time discretization introduces high-order diffusive and dispersive differential operators, the most common of which (the result of first-order upwind schemes LeVeque, 2002; Toro, 2009) is the so-called "false diffusion"

[1]One notable exception is that of higher-order finite-difference schemes such as the PENCIL code (sixth-order accurate in space), the performance of which is essentially independent of the geometry of the mesh (Lyra et al., 2008)

(or "numerical viscosity" or "advection error"). This diffusion depends on the fluid velocity and the grid spacing and it is also a function of angle between the flow direction and the coordinate axis used to discretize the domain (de Vahl Davis & Mallinson, 1972; Patankar, 1980).

This property of Eulerian codes becomes specially relevant in astrophysical fluid dynamics, where the evolution of the energy equations is often important. In particular, cosmological simulations with advection-dominated flows (e.g. in extremely supersonic motion near the cosmological density peaks) can be particularly sensitive to the numerical scheme being used. In high-velocity flows, the kinetic energy density is much larger than the internal/thermal energy, and thus small fractional errors in the velocity can translate into large fractional error in temperature, eventually distorting the thermodynamic evolution of the gas. This has been referred to as the "high Mach number problem" (Ryu et al., 1993; Bryan et al., 1995; Feng et al., 2004; Trac & Pen, 2004). The grid-dependent diffusion of Eulerian codes explains why a cylindrical-coordinate discretization of the domain is the preferred choice to study accretion and protoplanetary disks, since the direction of the flow is almost purely azimuthal The high-Mach-number problem can be alleviated in cylindrical coordinates by the use of the FARGO[2] technique (Masset, 2000), which removes the Keplerian speed at a given radius at the moment of numerically solving the Euler equations, thus effectively using a non-inertial moving frame in which the velocity changes are small. However, thin accretion disks are not highly supersonic in the azimuthal direction (the orbital Mach number is $\mathcal{M} \sim 1/h \sim 50$ for aspect ratios of 0.02), but they can also have supersonic bulk speeds. For young stellar associations,

[2]The most obvious benefit of implementing FARGO () is the great increase in the allowed time-step size, but the underlying benefit is reducing the numerical diffusion by ignoring the bulk velocity of the flow and instead solving for the deviations from it.

pre-main sequence stars can move at relative speeds of 1-3 km s^{-1}, which is roughly ten times or more the sound speed of molecular gas at a temperature of 10 K.

If there is a strong deviation from a point mass potential (e.g. owing to the presence of a large mass ratio binary) or if this point mass is allowed to move across the computational domain, the flow is no longer one-directional in this coordinate system, and the choice of a cylindrical geometry is no longer the most obvious one. In cases where there is no obvious symmetry that can be exploited through a suitable choice of a coordinate frame, numerical studies commonly use either adaptive mesh refinement (AMR; Berger & Oliger, 1984; Berger & Colella, 1989) on a cartesian grid, or smoothed particles hydrodynamics (SPH; Lucy, 1977; Gingold & Monaghan, 1977; Monaghan, 1992; Springel, 2010b). AMR codes used for star formation simulations such as ORION (e.g. Truelove et al., 1998) have been used successfully to simulate circumstellar disk formation over a several orbital timescales (Kratter et al., 2010). Thanks to mesh refinement, these codes can achieve very high resolution, although the levels of refinement are usually a fixed number, and the dynamical range in density that is achieved is smaller than that of particle-based codes. Similarly, although the higher resolution minimizes the effects of the high-mach-number problem, the grid is still subject to such limitation. On the other hand, SPH – a very popular tool for self-gravitating astrophysical fluid dynamics – offers low-order accuracy for the treatment of contact discontinuities and has poor shock resolution. In addition, SPH appears to suppress fluid instabilities under certain conditions (Agertz et al., 2007), suffer from subsonic velocity noise (Abel, 2011), and is known to require at least a factor of two more resolution elements to achieve similar levels of accuracy in simulations of quiet circumstellar disks (de Val-Borro et al., 2006).

One alternative is the moving-mesh approach of (Springel, 2010a) (see also Borgers & Peskin, 1987; Trease, 1988; Dukowicz et al., 1989, for earlier approaches). In this approach, the unstructured grid makes the geometry of the problem irrelevant for the discretization of the equations of hydrodynamics. This moving-mesh method is a finite-volume, Godunov, MUSCL-Hancock scheme with piece-wise linear reconstruction, thus being second-order accurate in space and time. The non-linear step of the MUSCL-Hancock technique (i.e. the solution of the Riemann problem) is carried out in the moving-frame (the instantaneously-at-rest frame) of each cell interface. This provides a method in which a gas disk around a young star can be modeled and evolved independently of the motion of said star, whether it is at rest, moving in a straight line, or orbiting another star: in each of these situations, the numerical scheme would be unaltered and the mesh would evolve according to the motion of the gas.

5.2 Numerical Method

AREPO is a second-order Godunov, finite-volume, moving-mesh scheme. The mesh is constructed from the Voronoi tessellation of a set of generating points that move with the local velocity of the flow. The primitive variables are reconstructed to linear order using slope-limited gradients (MUSCL reconstruction), which are obtained from all neighboring cell-centered primitive variables by means of the Green-Gauss theorem. The time-marching scheme consists of a predictor-corrector method which uses a linear spatial extrapolation to the center of each interface, followed by the time evolution of the primitive variables for a half time-step (MUSCL-Hancock method). The time-centered Godunov fluxes are obtained by solving a local 1-D Riemann problem in the *rest frame of*

the moving interface. If an exact Riemann solver is used, the sampled solution provides time-centered primitive variables at the center of the face; the velocity is boosted back to the lab frame; finally, the Godunov fluxes are obtained from the analytic form of the flux using the numerically obtained primitive variables. If an approximate-state solver is used for the Riemann problem, this is implemented in lab-frame coordinates (but with axes aligned with the local face); the Godunov fluxes result directly from the approximate solver, and an advection-flux term needs to be subtracted from these in order to take the motion of the mesh into account.

Gravitational potentials and gas self-gravity are computed using an N-body gravity tree (Barnes & Hut, 1986), where the tree leaves are the gas cells, whose contribution to the gravitational potential uses an adaptive softening length that is proportional to the cell radius (Springel, 2010a).

5.3 Isolated Thin Disk Models

In this work, we model circumstellar thin disks in equilibrium by using direct numerical simulation. Stationary, axisymmetric models must satisfy the equilibrium equations

$$-\frac{v_\phi^2}{R} = -\frac{1}{\rho}\frac{\partial p}{\partial R} - \frac{\partial \Phi_0}{\partial R} - \frac{\partial \Phi_g}{\partial R} \tag{5.1}$$

$$0 = -\frac{1}{\rho}\frac{\partial p}{\partial z} - \frac{\partial \Phi_0}{\partial z} - \frac{\partial \Phi_g}{\partial z} \tag{5.2}$$

(i.e. the equations of radial centrifugal equilibrium and vertical hydrostatic equilibrium respectively) where $\Phi_0(R, z)$ is the potential due to the central star and $\Phi_g(R, z)$ is the potential due to gas self-gravity. Equation 5.1 will help us determine the azimuthal velocity field $v_\phi^2(R, z)$ while Equation 5.2 will help us solve for the vertical structure of

the disk at all radii.

5.3.1 Model Characteristics

In the models presented here, we make use of the Lynden-Bell-Pringle density profile (Lynden-Bell & Pringle, 1974):

$$\Sigma(R) = (2-p)\frac{M_d}{2\pi R_c^2}\left(\frac{R}{R_c}\right)^{-p}\exp\left[-\left(\frac{R}{R_c}\right)^{2-p}\right] \tag{5.3}$$

where M_d is the total disk mass, R_c is the disk's characteristic radius and p is a surface density power-law index. Such a surface density profile is not just theoretically motivated, but also consistent with observations (Hughes et al., 2008; Andrews et al., 2009). In addition, it is a computationally convenient setup, since smoothly tapered disks are quieter/more stable ICs than truncated power-law disks, that show impulsively started rarefaction "shocks" in the outskirts due to discontinuous pressure gradients that fall sharply to zero.

We also impose a fixed temperature profile of the form:

$$T(R) = T_c\left(\frac{R}{R_c}\right)^{-l} \tag{5.4}$$

with the density and temperature power-law indices fixed to $p = 1.0$ and $l = 0.5$, respectively, in consistency with circumstellar disk structre derived from dust-continuum observations (Andrews et al., 2009). In addition, we fix the disk characteristic radius to $R_c = 20$ AU and the total mass of the system (star plus disk) to 0.5 $M_\odot$. The disk-to-star mass ratio is varied from $M_d/M_* = 0.02$ up to 0.67, always keeping $M_* + M_d = 0.5\ M_\odot$. The normalization of the temperature profile in Equation 5.4 is chosen according to a

specified aspect ratio at R_c. The nominal aspect ratio of the disk is

$$h(R) \equiv \frac{c_s(R)}{v_K(R)} = \sqrt{\frac{k_B T(R)}{\mu m_{\rm H}} \frac{R}{GM_*}} \equiv h_c \left(\frac{R}{R_c}\right)^{-(l-1)/2} \tag{5.5}$$

i.e., that of a flared disk. The normalization factor h_c was chosen from the values 0.04, 0.06, 0.08 and 0.1. A very thin disk of $h_c = 0.02$ (which implies h of less than 0.01 at 1 AU) is challenging for a proper three-dimensional description unless high-resolution ($N_{\rm gas} \gtrsim 10^6$) models are used.

All singular terms in Equations 6.1 - 5.5 are regularized using a spline softening with softening parameter h of the same kind to the one used for the gravitational potential of the central stars and of individual gas cells (Springel et al., 2001). In this case, the central gravitational potential Φ_0 reaches a finite value at $R = 0$ (the position of the star) and recovers its exact Keplerian value at $R = 2.8h$. This means that the Keplerian angular speed $\Omega_K^2 = (\partial\Phi_0/\partial R)/R$ is small at $R = 0$. The resulting surface density profiles is as shown in Figure 5.1, where we have varied the total disk mass (M_d in Equation 6.1). Figure 5.2 shows the Toomre Q parameter as a function of radius for disks with different masses and temperature scalings.

The one-dimensional parametrization of Equation 6.1 of an axisymmetric disk is sufficient to numerically determine a three-dimensional disk in hydrostatic and centrifugal equilibrium. The generation of such initial conditions discretized into a Voronoi-tessellation mesh is described below.

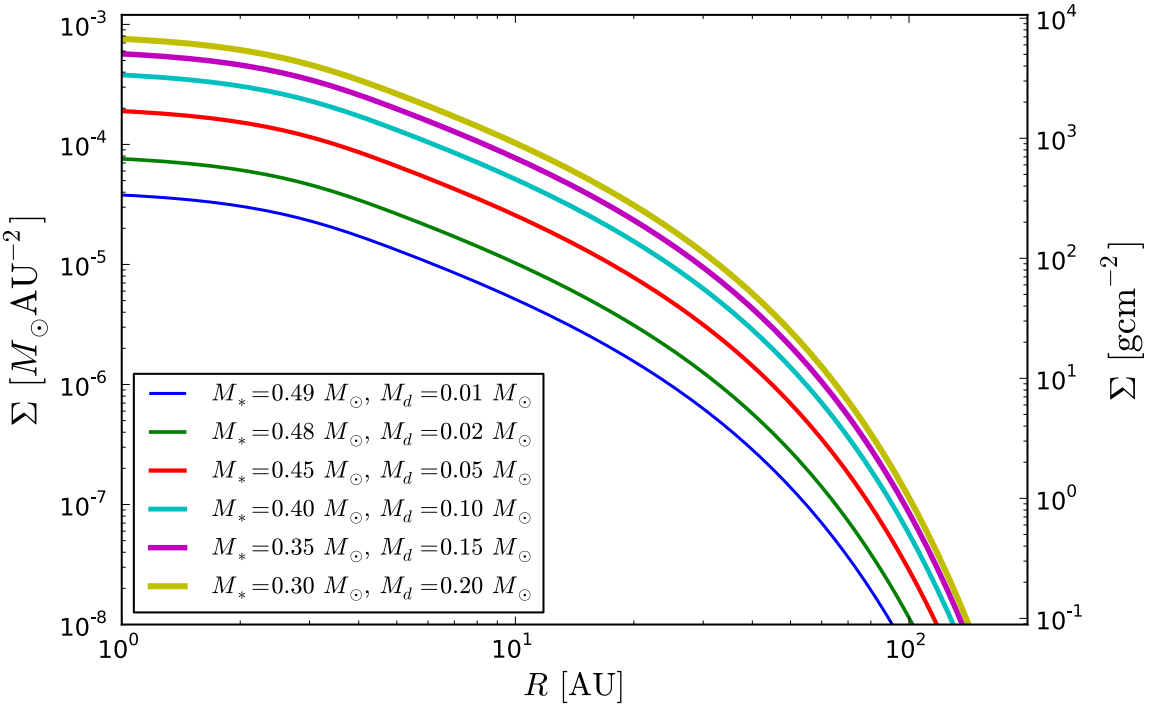


Figure 5.1.— Lynden-Bell-Pringle surface density profiles (Equation 6.1) for different combinations of star and disk masses while keeping the system mass constant ($M_* + M_d = 0.5 M_\odot$).

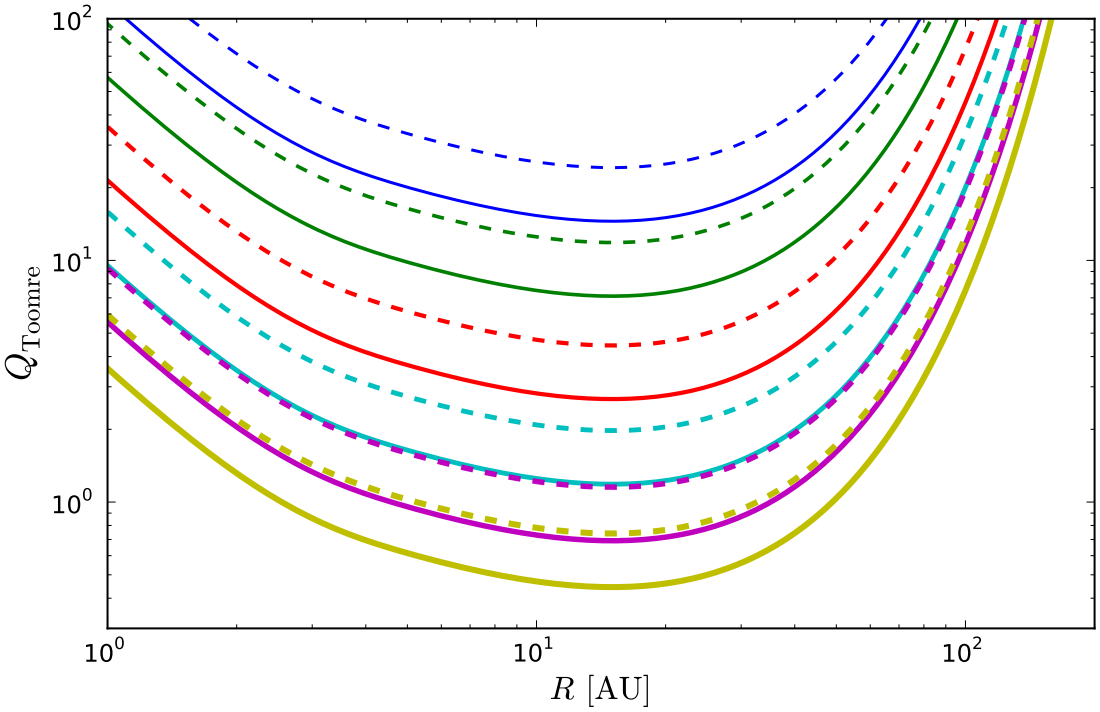


Figure 5.2.— Toomre's Q-parameter as a function of radius for different model setups. Solid lines correspond to disks with a fixed h_c (Equation 5.5) of 0.04 (i.e., colder disks), while dashed lines correspond to profiles with $h_c = 0.06$ (warmer disks). The different colors correspond to the different disk masses as described in Figure 5.1. Solid lines (from top to bottom): $Q_{\rm min} = 14.5, 7.1, 2.7, 1.2, 0.7$ and 0.4. Dashed lines (from top to bottom): $Q_{\rm min} = 24.2, 11.0, 4.4, 2.0, 1.2$ and 0.7.

5.3.2 Initial Conditions

Mesh Generation

Structured-grid codes commonly do this following a space-based criterion, in which cells have a constant volume (or a hierarchy of fixed volumes in AMR) and the mass of each cell varies according to conservation of mass and momentum. Pseduo-Lagrangian approaches like SPH, discretize the computational domain following a mass-based approach, in which fluid particles represent (usually constant-mass) parcels of fluid that have variable effective volumes (softening kernels). As a quasi-Lagrangian mesh code, AREPO can discretize the domain initially following either space-based or mass-based approaches, or a combination of both. In this work, we create circumstellar disk models with nearly constant mass cells (i.e. with variable volume) immersed in a background computational box that is discretized by nearly constant-volume cells (Springel, 2010a).

In a tessellation-based code, the structure of the mesh is entirely and uniquely determined by the positions of the mesh-generating points (which are, in turn, proxies to gas "particles"). In the present work, we choose to to aim for a given *mass resolution* instead of a spatially-based discretization of the domain. This is accomplished in a similar manner to pseudo-Lagrangian, particle-based codes, in which the domain is discretized by a Monte-Carlo sample of an underlying density field.

Given a number of mesh-generating points $N_{\rm gas}$, a gas disk of total mass M_d is discretized into cells of nearly equal mass $m_{\rm gas} = M_d/N_{\rm gas}$. A three-dimensional axisymmetric model requires the specification of the cell coordinates x, y and z, or equivalently, of R, z and ϕ, the last of which is drawn from a uniform distribution in

$[0, 2\pi)$ while the former two must reproduce the continuous density field $\rho_{\rm gas}(R, z)$.

In its simplest form, Monte-Carlo sampling entails inverting some 1-D cumulative mass function to solve for the position of a gas element in coordinate space given a randomly sampled mass fraction $[0, 1) \times M_{\rm total}$. If some 3-D system (that is not spherically symmetric) cannot be separated into three independent cumulative mass functions along the three coordinate axis, the Monte-Carlo sampling of the density field can be technically difficult. In the case of circumstellar disks, we assume an axisymmetric 3-D density field of the form $\rho(R, z) = \Sigma(R)\zeta(R, z)$, where $\int_{-\infty}^{+\infty} \zeta(R, z)dz = 1$. In this case, where the density field is not separable into R and z components.

The single variable function $\Sigma(R)$ is our arbitrary surface-density model of Equation 6.1. This surface density can be sampled directly using the procedure described above, producing a list of $N_{\rm gas}$ radial positions $\{R_i\}$.Then, assuming the variability in R is slower than that in z, we group the different $\{R_i\}$ into radial bins and proceed to Monte-Carlo sample the 1-D function $\zeta(z|R)$ to obtain values for the z-coordinate. For this, we need to have a solution for the vertical profile $\zeta(z|R)$ first, which, unless disk self-gravity is entirely ignored, can only be obtained numerically.

Vertical density structure

Specifying the surface density profile is only the first step in constructing stationary ICs. In three dimensions, one has to solve for the vertical structure of the disk self-consistently if gas self-gravity is of any importance.

For the vertical structure, we use the potential method described in detail in Wang et al. (2010a). This techniques consists of iterating the between vertical hydrostatic

equilibrium equation – for a fixed vertical potential – and the vertical Poisson equation – for a fixed vertical density profile – until convergence is achieved to within a tolerance parameter (typically anywhere between 10^{-8} and 10^{-12}).

The geometrically thin disk approximation simplifies calculations significantly, since it allows us to solve for two coupled ordinary differential equations instead of a set of partial differential equations. The first approximation involves forcing a separability of the potential of a very flattened system into a mid-plane component and a local vertical component (Binney & Tremaine, 2008)

$$\Phi_g(R,z) = \Phi_g(R,0) + \Phi_{g,z}(R,z)$$

such that the Poisson equation for a given R can be written

$$\frac{\partial^2 \Phi_g(R,z)}{\partial z^2} = \frac{\partial^2 \Phi_{g,z}(R,z)}{\partial z^2} = 4\pi G\rho(R,z) \quad , \tag{5.6}$$

The second approximation is to assume that the sound speed only depends on the radial coordinate R. Therefore, if $p = c_s^2(R)\rho$ at a given R, Equation 5.2 is simply

$$\frac{c_s^2(R)}{\rho}\frac{\partial \rho}{\partial z} = -\frac{\partial \Phi_0}{\partial z} - \frac{\partial \Phi_g}{\partial z} = -\frac{\partial \Phi_{0,z}}{\partial z} - \frac{\partial \Phi_{g,z}}{\partial z}$$

where $\Phi_{0,z} \equiv \Phi_0(R,z) - \Phi_0(0,z)$ is the z-dependent part of the Keplerian potential Φ_0 of the central star.

Thus, provided we know the mid-plane density $\rho(R,0) \equiv \rho_0(R)$, we can always solve for the vertical density profile

$$\rho(R,z) = \rho_0(R)\ \exp\left(-\frac{\Phi_{0,z} + \Phi_{g,z}}{c_s^2(R)}\right) \quad . \tag{5.7}$$

Given an initial mid-plane guess $\rho_0^{(0)}(R)$, we solve for an initial vertical profile $\rho^{(k)}(z)$ (Equation 5.7), and then update successively the quantity $\rho_0^{(k)}(R)$ until a satisfactory

solution for $\rho(R,z) = \Sigma(R)\zeta(z|R)$ is obtained. The iteration steps are (for fixed R):

$$\text{(I)} \quad \rho_0^{(k+1)} = \frac{\Sigma(R)}{\int_{-\infty}^{\infty} \rho^{(k)}(z)\, dz} \tag{5.8a}$$

$$\text{(II)} \quad \text{solve numerically for } \Phi_{g,z}^{(k+1)}$$

$$\frac{d}{dz}\begin{pmatrix} \Phi_{g,z}^{(k+1)'} \\ \\ \Phi_{g,z}^{(k+1)} \end{pmatrix} = \begin{pmatrix} 4\pi G\rho^{(k)}(z) \\ \\ \Phi_{g,z}^{(k+1)'} \end{pmatrix} \tag{5.8b}$$

$$\text{(III)} \quad \rho^{(k+1)}(z) = \rho_0^{(k+1)} \exp\left(-\frac{\Phi_{0,z} + \Phi_{g,z}^{(k+1)}}{c_s^2}\right) \quad , \tag{5.8c}$$

where the initial guess for the disk self-potential $\Phi_{g,z}^{(0)}$ is zero.

Once a numerically obtained vertical profile for fixed R is available for a list of values of z, we proceed to assign z-coordinates to all the sampled points that fall into the bin of radius R. This is done in analogous manner to the sampling of the R-coordinate given $\Sigma(R)$, using Monte Carlo sampling.

The convergence speed depends in great part on how good the initial guess is. For low mass circumstellar disks ($M_d \lesssim 0.05 M_*$) convergence can be achieved within at most 5-6 iterations if an initial mid-plane density corresponds to that of a Gaussian-profle disk (the exact profile for non-self-gravitating disks). If disks are massive ($M_{\gtrsim} 0.1 M_*$), convergence can take up to ~ 300 iterations if the initially Gaussian profile is used. Instead, the self-gravitating slab (Ledoux, 1951) is a much better initial guess (although it neglects the presence of the star's Keplerian potential). In this case, convergence can be reached with at most ~ 30 iterations. For disks masses of $M_d \gtrsim 0.2 M_*$, the latter approach needs at most ~ 9 iterations for convergence.

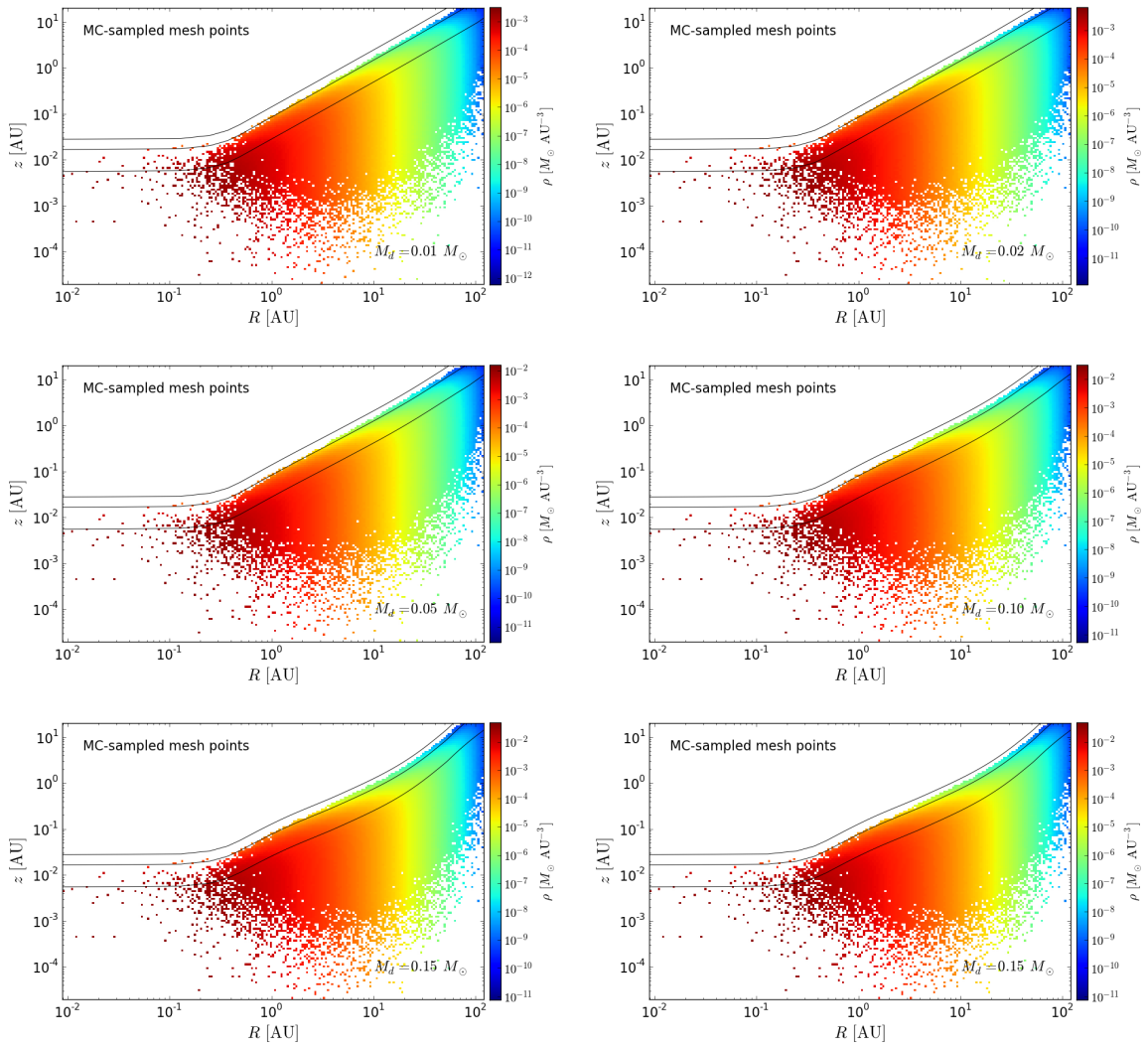


Figure 5.3.— Density field $\rho(R, z)$ obtained by numerical solution of the vertical disk structure for six different disk masses $M_d = 0.01, 0.02, 0.05, 0.1, 0.15$ and $0.2\,M_\odot$. All surface density profiles follow the parametric form of Equation 6.1 with fixed power-law index $p = 1$ and identical temperature profiles ($l = 0.5$). The three solid black lines represent 1, 3 and 5 times the disk scale height, which is defined as $H(R) \equiv [\int \rho(R, z) z^2\, dz / \int \rho(R, z)\, dz]^{1/2}$

This process gives a solution for $\rho(R, z)$ within an arbitrary error tolerance. The two-dimensional density field is shown in Figure 5.3 for different disk-to-star mass ratios, showing how the vertical structure is modified as more mass is added to the disk. The evaluation of the density field is complete in the vertical direction up to 3 scale-heights, where it is truncated due to the finite number of cells drawn from the underlying density distribution for which we have solved. The larger $N_{\rm gas}$ (i.e. the smaller $m_{\rm gas}$) the further the density field is sampled into the z-direction. At the "last-cell-limit", one must transition into a quasi-regularly spaced mesh that can make the transition into the background mesh that fills the computational domain (see Springel, 2010a for a detailed description). Obtaining a smooth transition is difficult to accomplish, in particular for circumstellar disks. Owing to disk flaring, the transition into the background mesh at small distances from the star happens at values of z much smaller than at large distances from the star. Iterative relaxation steps such as the Lloyd algorithm (Springel, 2010a) can be used to smooth out distorted cells located in this transition. Figure 5.4 shows the same solutions to the density field as Figure 5.3, but in this case the density field has been evaluated further into the vertical direction (beyond the limit imposed by the finite-size random sample), where low-mass cells have been added to allow for a transition into the very dilute background. Figure 5.7 shows a *slice* through the mesh for one if the isolated disk models. The dense region of the mesh maps the disk at nearly constant mass per cell, while the low resolution background discretized the computational domain at nearly constant volume per cell.

Figures 5.4 and 5.3 show how the disk self-gravity affects the shape of the disk in the vertical direction. For low mass disks – where the vertical profile is very close to Gaussian – the different scale-height levels show constant slopes in logarithmic coordinates. The

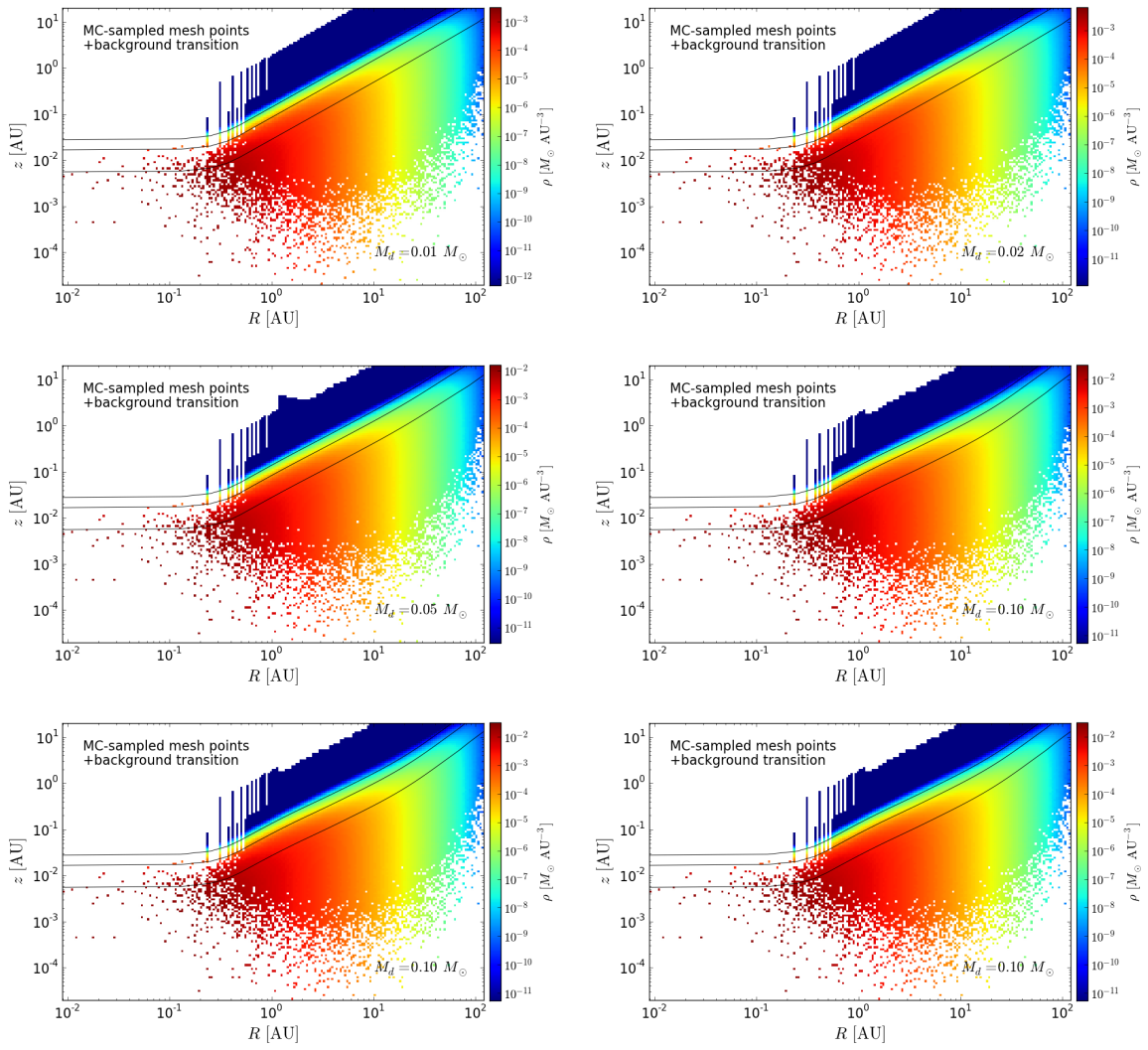


Figure 5.4.— Same as Figure 5.4 but including the density field assigned to the additional transition mesh above the region of the disk sampled by the Monte-Carlo technique. Before a density floor is reached, the disk density field is completely sampled up to five scaleheights.

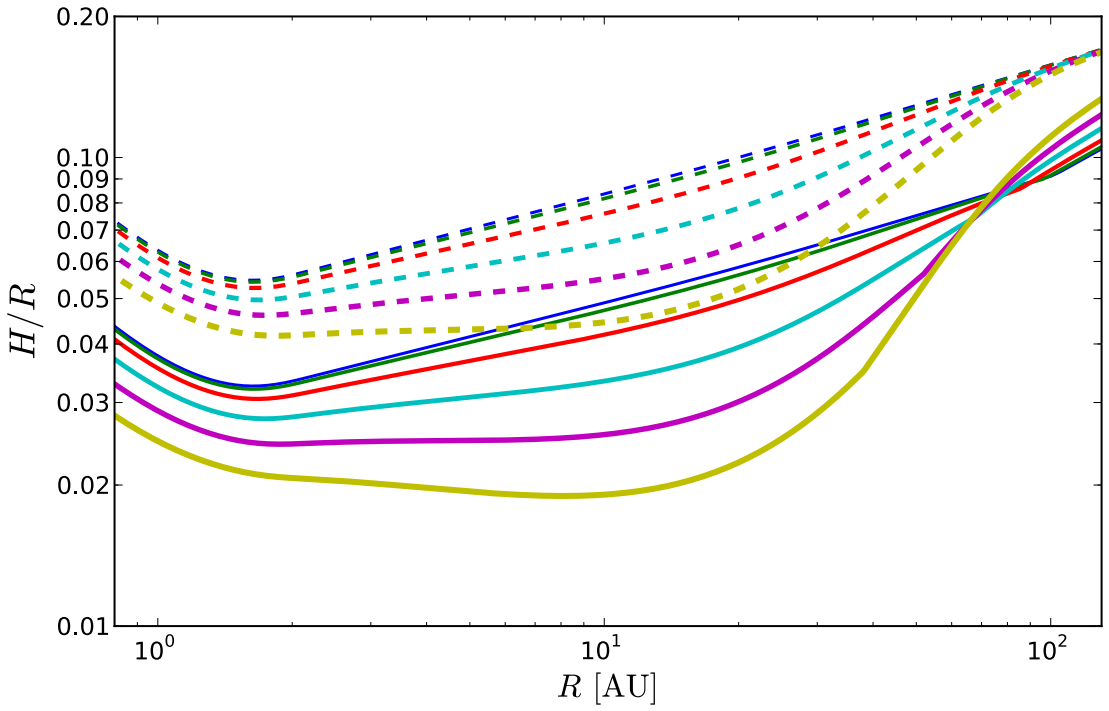


Figure 5.5.— Numerically computed scale-height profiles for the six different surface density profiles. Line color and line thickness are the same as in Figures 5.1 and 5.2

slopes of these curves is given by $H(R) \sim c_s/\Omega_K \propto R^{(3-l)/2} \propto R$. At larger masses, however, the scale-height, defined at each R as $H(R) \equiv [\int \rho(R,z) z^2\, dz / \int \rho(R,z)\, dz]^{1/2}$ is no longer described by a power-law. This effect is better illustrated by the aspect ratio profiles shown in Figure 5.5. The aspect ratio $h = H/R$ is greatly affected by self-gravity, becoming even non-monotonic for large disk masses. This mid-plane concentration disappears rapidly at $\sim$ 60 AU $\sim 3R_c$, where the most massive disks flare up to match the aspect ratio of the low mass disk. This effect is a direct consequence of the thin-disk approximation and the locality of the vertical Poisson equation (Equation 5.6); once the local surface density is low enough, the disk vertical structure responds to the gravitational potential of the enclosed mass ($\sim M_* + M_d(< R) \sim 0.5\ M_\odot$), thus approaching non-self-gravitating limit in the outer regions regardless of the total disk mass. Since in our models $M_* + M_d = 0.5 M_\odot$ for all disk-to-star mass ratios, the aspect ratio in the outer regions is nearly the same regardless of the scaling parameter h_c, since far away from the central star, gas temperature is not allowed to fall below 10 K.

The Monte-Carlo process is intrinsically stochastic, and one of its known consequences is the Poisson fluctuations in the resulting density field. Such a "noise" can only be overcome by increasing the number of particles that represent the underlying continuum field. However, to minimize the Poisson noise in AREPO, we take advantage of its Eulerian nature by imposing the exact value of the density field at the mesh-generating points, instead of forcing each cell mass to have the same value. The location of those points, nevertheless, is still specified by the Monte-Carlo sampling of the density field. This additional step – only possible in AREPO – produces very smooth density fields, at the expense of a variation in cell mass. However, this choice relies on the accuracy of the linear density reconstruction within each cell (which allows for the conversion

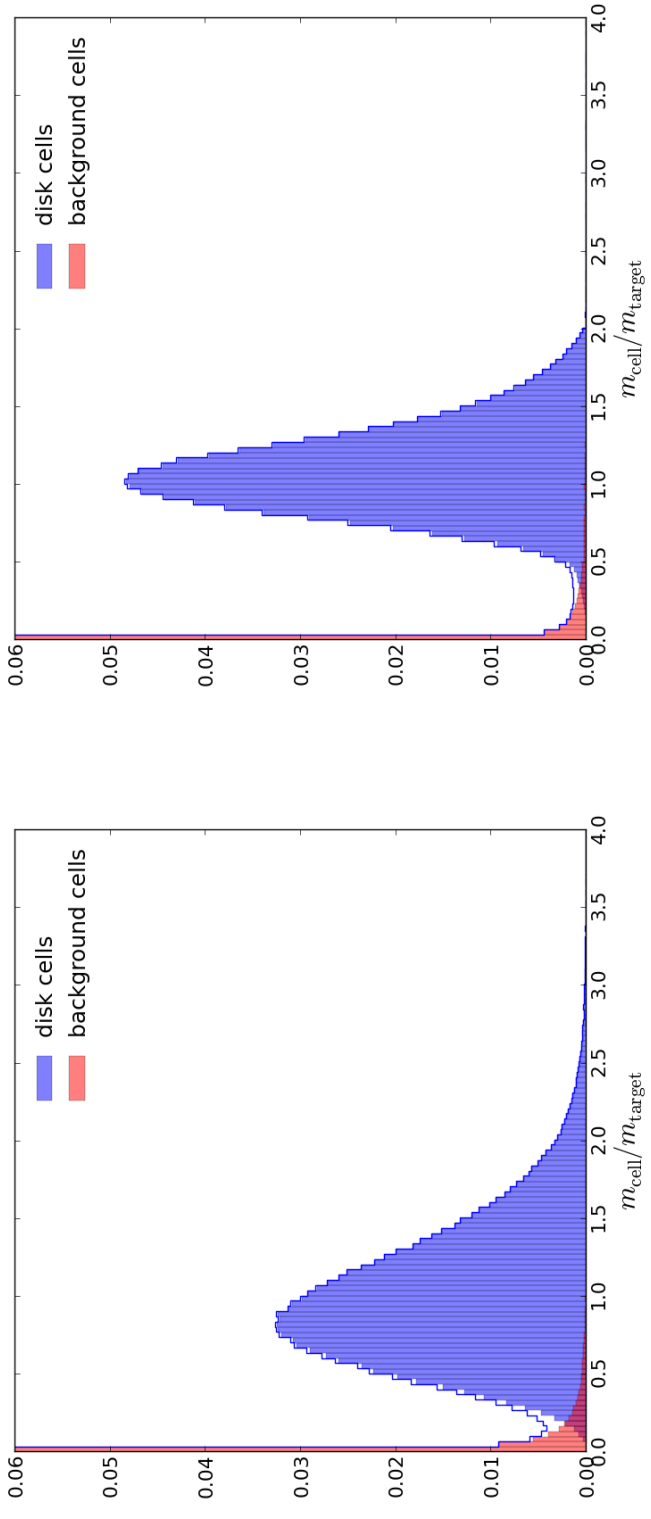


Figure 5.6.— Mass distribution of Voronoi cells resulting from the tessellation of Monte-Carlo sampled points (left panel) and the distribution resulting from refinement/derefinement of cells above/below the mass limit 0.5/2.0 times the reference mass m_{target} (right panel).

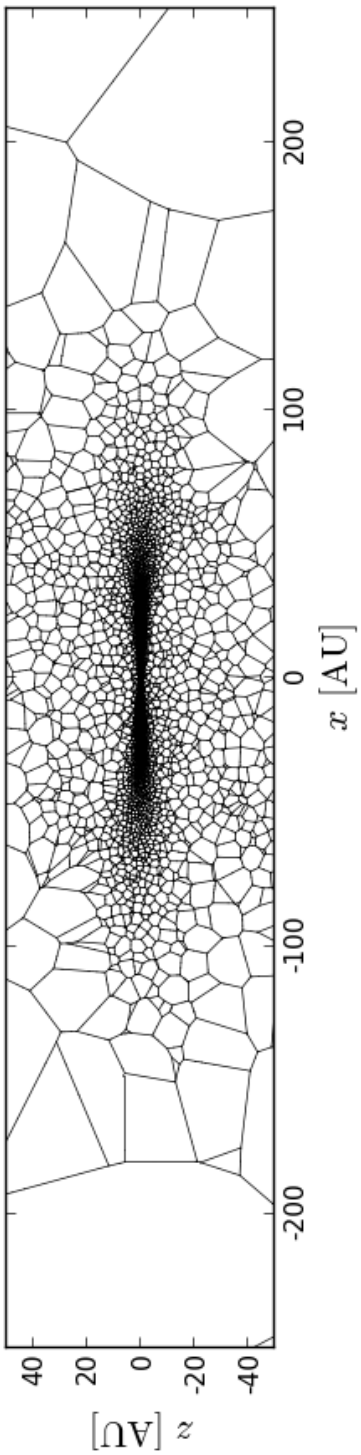


Figure 5.7.— Slice through the mesh of a disk with $N_{\rm gas}$ = cells, showing the transition between a mass-based sampling within the disk and a space-based sampling in the background.

$\rho_{\rm cell} = m_{\rm cell}/V_{\rm cell}$). In regions of steep density gradients, linear reconstruction will be a too rough of an approximation, and errors in the total mass of the cell will be introduced. In practice, the distribution of cell mass will have some spread around the desired cell mass $m_{\rm target} = m_{\rm gas}$. Therefore, this method represents a trade-off between Poisson noise in the density field, and a spread in the mass-per-cell distribution. This spread is of little concern because (a) the very nature of the Godunov scheme will make strict mass constancy impossible at later times, and (b) refinement/derefinement will immediately correct those pathological cases where a few cells have highly discrepant masses. Figure 5.6 shows the distribution of cell masses as generated by the IC algorithm and how the spread in cell masses can be corrected by a series of refinement and derefinement steps (Springel, 2010a).

Figure 5.6 also shows the distribution of "background cells", which are very low mass cells or large volume that fill in the computational box. These cells which are not subject to the refinment/derefinement regularization of the main part of the mesh, and thus are allowed to have masses much below the reference mass $m_{\rm target}$. Ideally, these two cell distributions should smoothly transition into each other. Figure 5.7 show such transition between the dense concentration of cells of nearly constant mass where the disk lies and a diffuse background of large cells.

Finally, we confirm that the prescribed surface density $\Sigma(R)$ is recovered from the 3D models by integrating the density field along the vertical direction. We generate images (Figure 5.8) for an $M_d = 0.05\ M_d$ disk on the x-y and x-z planes for four different temperature normalizations or reference aspect ratios: $h_{R_d} = 0.02, 0.04, 0.06$ and 0.1.

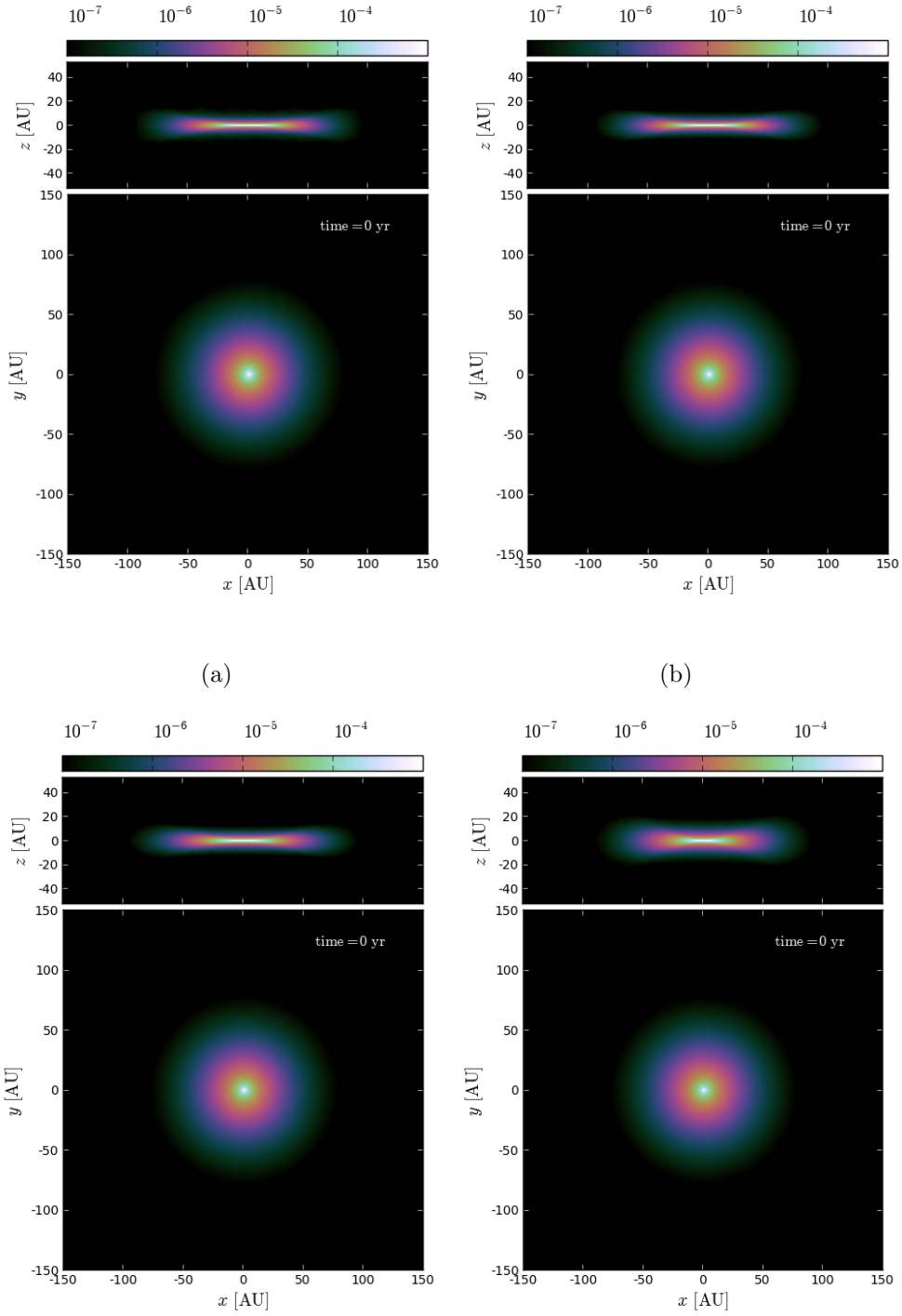


Figure 5.8.— Different initial conditions for disk with $M_d = 0.02\,M_\odot$ orbiting around a star of of mass $0.48\,M_\odot$. From left to right, the temperature of the disk is increased such that the scale-height at $R = R_c$ equals (a) 0.02, (b) 0.04, (c) 0.06 and (d) 0.1 .

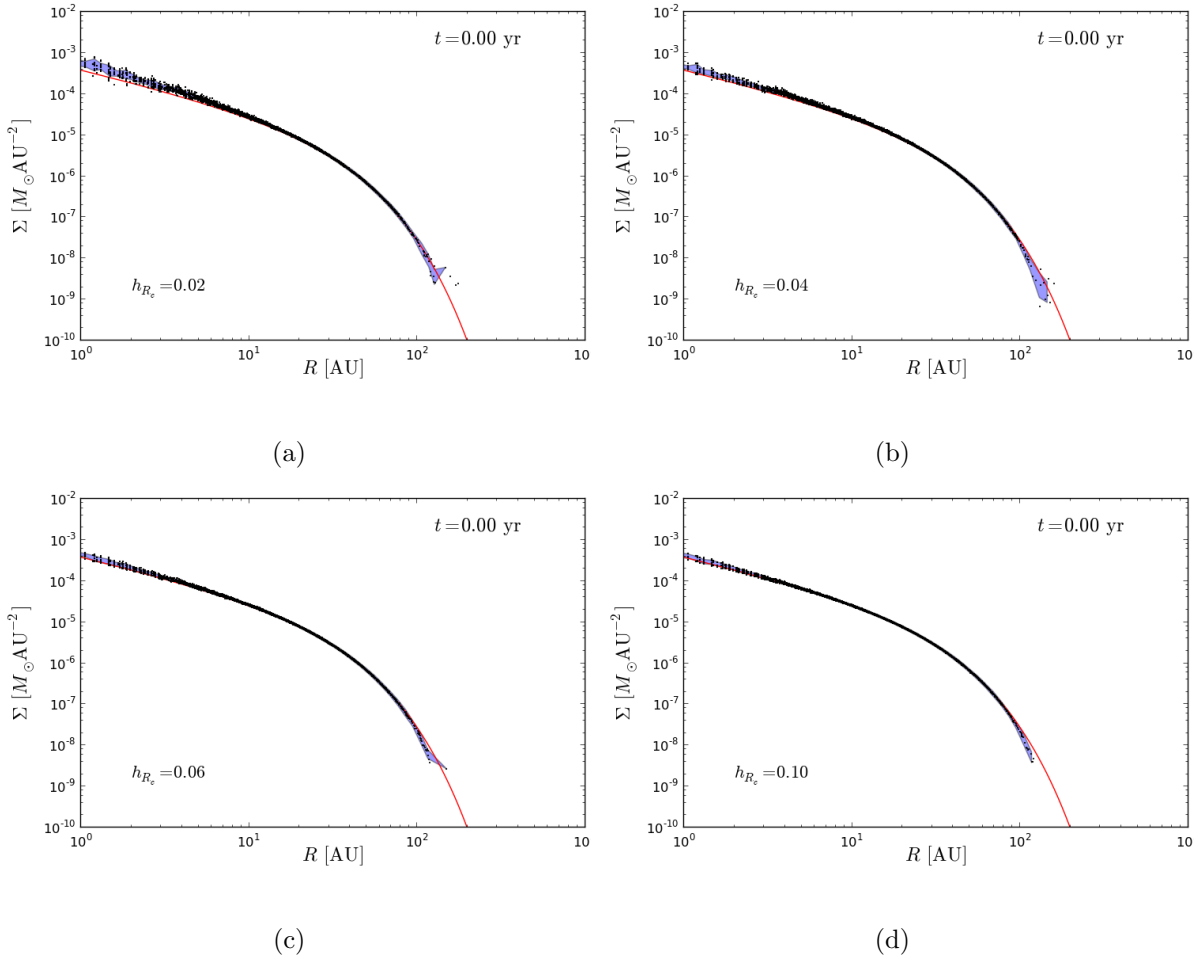


Figure 5.9.— Different initial conditions for disk with $M_d = 0.02\,M_\odot$ orbiting around a star of of mass $0.48\,M_\odot$. From left to right, and top to bottom, the temperature of the disk is increased such that the scale-height at $R = R_c$ equals $0.02, 0.04, 0.06$ and 0.1 respectively.

Velocity structure

The velocity field of a stationary, axisymmetric system is given by the solution for v_ϕ in Equation 5.1. To first order, the velocity profile of a low mass disk will be that of a Keplerian orbit. Secondary corrections are the negative contribution of the pressure gradient and disk self-gravity. In addition, three-dimensional disks will have a small vertical gradient in orbital speed due to which the upper layers rotate more slowly than the mid-plane (actually, the sign of this gradient depends on the sign of the radial temperature gradient; see below).

For low mass disks ($M_d/M_* < 0.01$) the pressure term is the most (and essentially only) modification to the Keplerian speed, amounting to a few percent of total speed. In the present work, we consider circumstellar disk models for which self-gravity is of some significance. In these models, ($M_d/M_* > 0.02$), the self-gravity term is comparable to, or greater than the pressure term, although it only starts to cause a significant deviation from Keplerian rotation for $M_d/M_* \gtrsim 0.3$.

Quantitatively, the full velocity field in three dimensions, for locally isothermal disks, is (see Wang et al., 2010a)

$$\frac{v_\phi^2}{R} = \frac{v_c^2}{R} - \frac{1}{\rho}\frac{\partial p}{\partial R}\bigg|_{z=0} - \frac{\partial c_s^2}{\partial R}\ln\left(\frac{\rho(R,z)}{\rho_0(R)}\right) \quad , \tag{5.9}$$

where $v_c^2 = v_K^2(z=0) + v_{c,d}^2$ is the circular speed due to gravity (from both the central star and the disk). Thus, vertically-layered rotation curve can be obtained by first calculating the two-dimensional rotation curve corresponding to a highly-flattened disk (Binney & Tremaine, 2008) and then adding a correction owed to the vertical structure (Wang et al., 2010a).

The different contributions to the velocity field can be written analytically if we ignore the effect of self-gravity on the vertical structure (i.e. the disk vertical aspect ratio is $h(R) = c_s/v_K \propto R^{(1-l)/2}$, Equation 5.5) as well as the singularity implied by the power-law forms of the temperature (Equations 5.4) and surface density (Equation 6.1). The velocity field is, to second order in z/R:

$$\begin{aligned} v_\phi^2 =& v_K^2 \left\{ 1 - h^2 \left[\frac{3}{2} + p + \frac{l}{2} \left(\frac{z^2}{h^2 R^2} + 1 \right) + \right. \right. \\ & \left. \left. (2-p)(R/R_c)^{2-p} \right] \right\} + v_{c,d}^2 \quad , \end{aligned} \tag{5.10}$$

where the only difference from the expression in Tanaka et al. (2002, Eq. 4) is the extra term due to the exponential tapering of the Lynden-Bell-Pringle surface density profile and the addition of the "circular speed" term due to disk self-gravity $v_{c,d}^2$.

The component of the circular speed do to self-gravity has many possible functional forms (see Binney & Tremaine, 2008). For numerical computations, we have found that a formula due to Mestel (1963) is particularly useful:

$$\begin{aligned} v_{c,d}^2(R) = G\frac{M_d(<R)}{R} + & \\ 2G \sum_{k=1}^{\infty} \alpha_k & \left[\frac{(2k+1)}{R^{2k+1}} \int_0^R dR'\, \Sigma(R') R'^{2k+1} \right. \\ & \left. - 2kR^{2k} \int_R^\infty dR'\, \frac{\Sigma(R')}{R'^{2k}} \right] \end{aligned} \tag{5.11}$$

Although the summation in Equation 5.11 does not converge particularly fast, it is' a convenient representation of $v_{c,d}^2$ for numerical computation, since it involves only well behaved integrals and simple sums. If a tolerance parameter magnitude of 10^{-6} is introduced relative to the zeroth order term ($GM_d(<R)/R$), only the first ~ 10 terms in the sum are necessary. In pathological cases where more terms are needed, the sum is

extended to ~ 20. This typically happens for small R, where often times the inaccuracy in $v_{c,d}^2$ is negligible relative to the dominant Keplerian term v_K^2.

For a density profile index of $p = 1$, the self-gravity contribution to the disk's rotation curve can be calculated exactly (see Appendix 5.5). Although the analytic solution to $v_{c,d}^2(R)$ is too complicated to be useful in practice (it involves Meijer functions), it can be used for debugging purposes and to compare to the numerical output of the truncated series in Equation 5.9. This solution can be written

$$v_{c,d}^2(R) = \frac{GM_d}{R_c} f\left(\frac{R}{R_c}\right) \tag{5.12}$$

where $f(x)$ is a dimensionless function[3] of order unity which involve Meijer-G functions (Appendix 5.5) and that equals 1 at $R = 0$ and approaches R_c/R as $R \to \infty$ (i.e. the disk potential approaches a Keplerian form at large distances). This is the velocity profile shown in Figure 5.11 next to the Keplerian and pressure terms.

However, the limitations of the analytic expression for $v_{c,d}^2$ are not only practical. Since $\Sigma(R)$ in Equation 6.1 diverges at the origin, the surface density there is infinite. The derivative of the gravitational field due to this mass distribution (Binney & Tremaine, 2008, Eq. 2.188) takes a finite value of GM_d/R_c at the origin, which is an unphysical value for the circular speed. Any continuous distribution of matter should have a vanishing circular speed at the origin, and this can be obtained by smoothing out the singular term in $\Sigma(R)$ as explained above. This produces a circular speed profile that is zero at the origin, but rapidly converges to the exact profile far from $R = 0$.

[3]The functional form of this dimensionles function is

$$f(x) = \frac{x}{2\sqrt{\pi}} G_{13}^{21}\left(\frac{x^2}{4}\,\middle|\, \begin{matrix} 0 \\ \frac{1}{2},\ -\frac{1}{2} \end{matrix} \right) \quad . \tag{5.13}$$

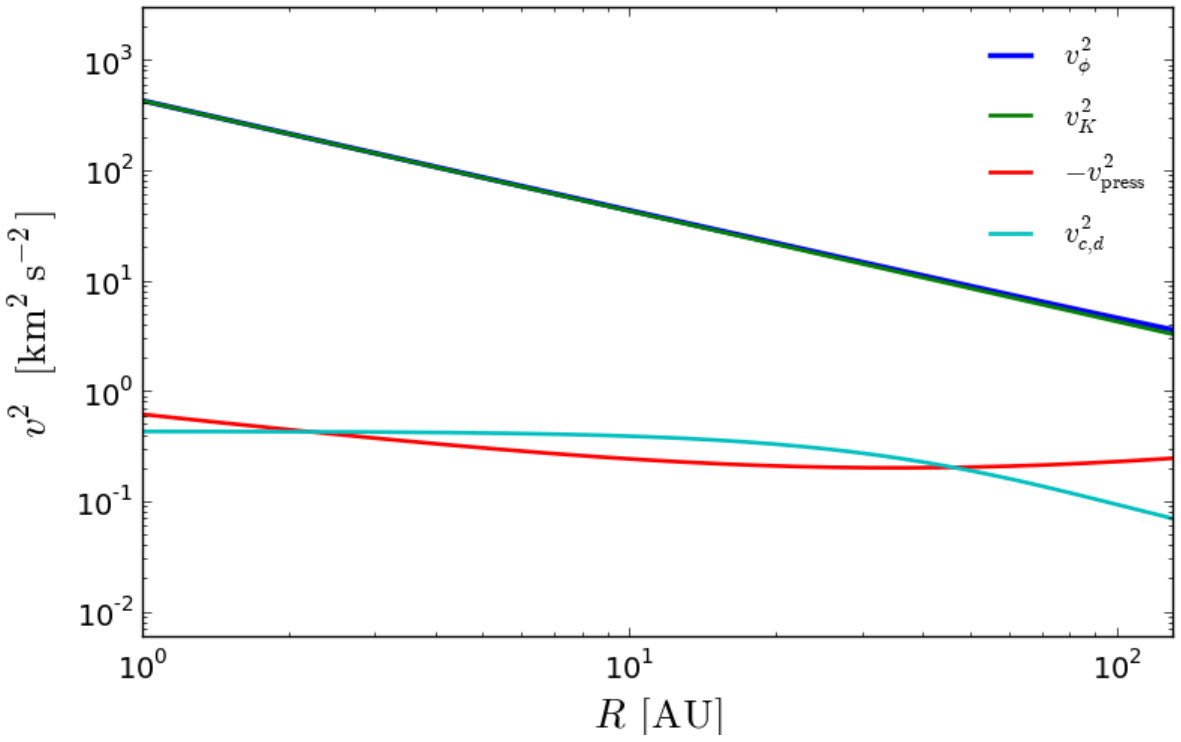


Figure 5.10.— Different components of the mid-plane velocity profile for a self-gravitating Lynden-Bell-Pringle disk.

This is shown in Figure 5.11, where the numerically obtained form of the circular speed (Equation 5.11) changes continuously from zero to the profile given by Equation 5.12.

In practice, all components of the velocity field are computed numerically to generate our disk models (for general p). The velocity components of all disk models listed in Figure 5.1 are shown in Figure 5.12. The model M0 (with $M_d = 0.01$ and $M_* = 0.49$) is qualitatively consistent, except for the smoothing features near the origin, with the analytic expressions of Figure 5.10.

5.3.3 Equation of state

A widely used approximation for the thermal structure of circumstellar disks is the so-called *locally isothermal approximation*. This approximation, arguably unphysical in the presence of shocks and rarefactions, aims to replicate a rather complicated radiative equilibrium of the gas with the radiation field of the star. Thin disks are difficult to treat with simplistic prescriptions for radiative transfer such as flux-limited diffusion. This is because the disk is very optically thick disk at the mid plane, while optically thin only a few scale heights above the plane, where its temperature is set by direct irradiation from the central star. For this set of runs we implement a quasi- locally isothermal approximation, in which the gas is assumed to be adiabatic, with an adiabatic index of $\gamma = 1.001$ (i.e. effectively isothermal) and cooled/heated at every time step in order to preserve the prescribed thermal energy $u(\mathbf{r}, \mathbf{r}_{*,1}, \mathbf{r}_{*,2})$, which is a function of position and the location of the stars. In practice, we limit ourselves to calculating the distance to the host star, and determine which sound speed to use based on that distance. The temperature profile is taken to be of the form $T \propto R^{-1/2} \approx r^{-1/2}$, where R is the

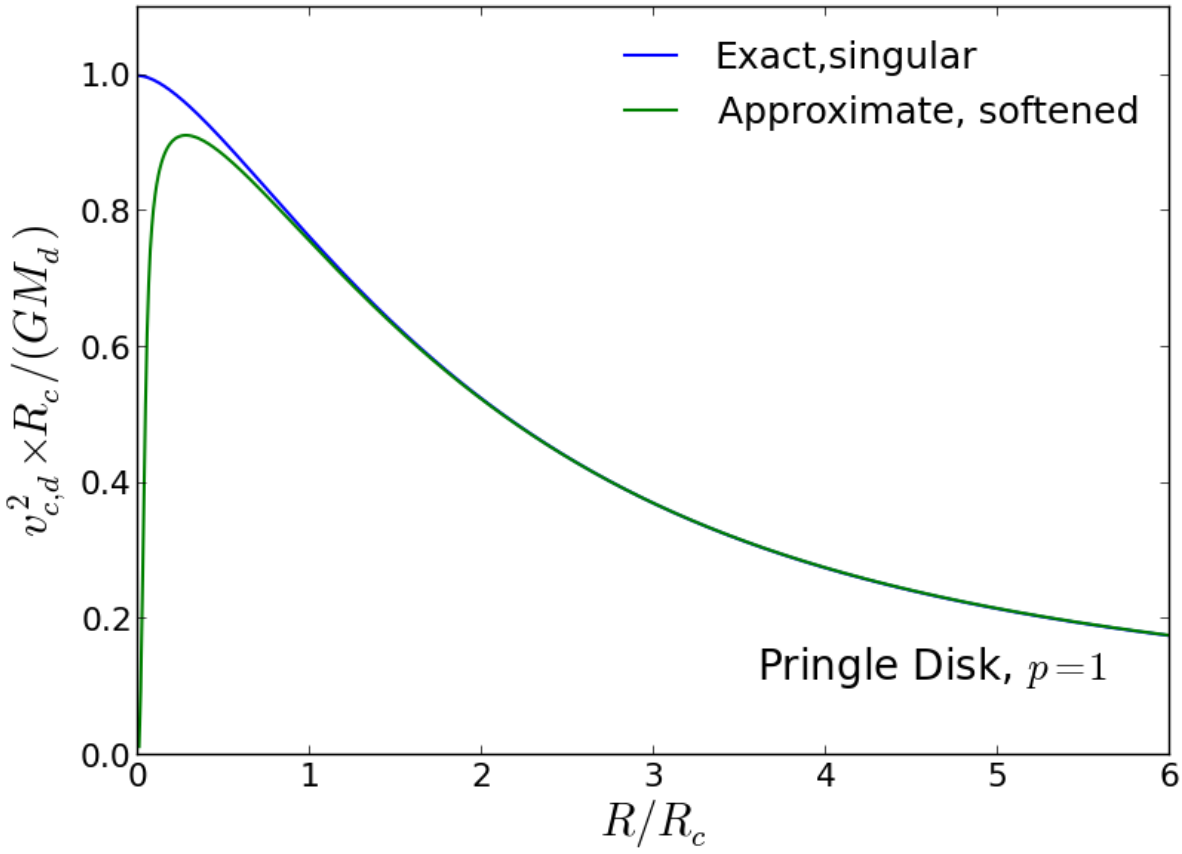


Figure 5.11.— Contribution to the circular speed profile of a Lynden-Bell-Pringle disk due to self-gravity. The blue line depicts the exact computation of $v_{c,d}^2$ (Equation 5.12) for the singular profile $\Sigma(R)$ of Equation 6.1 for a power-law index $p = 1$ normalized by GM_d/R_c. The green line shows the self-gravity contribution to v_ϕ we actually use in our models (calculated through a truncated version of Equation 5.11 by numerical integration using a version of $\Sigma(R)$ softened at the origin.

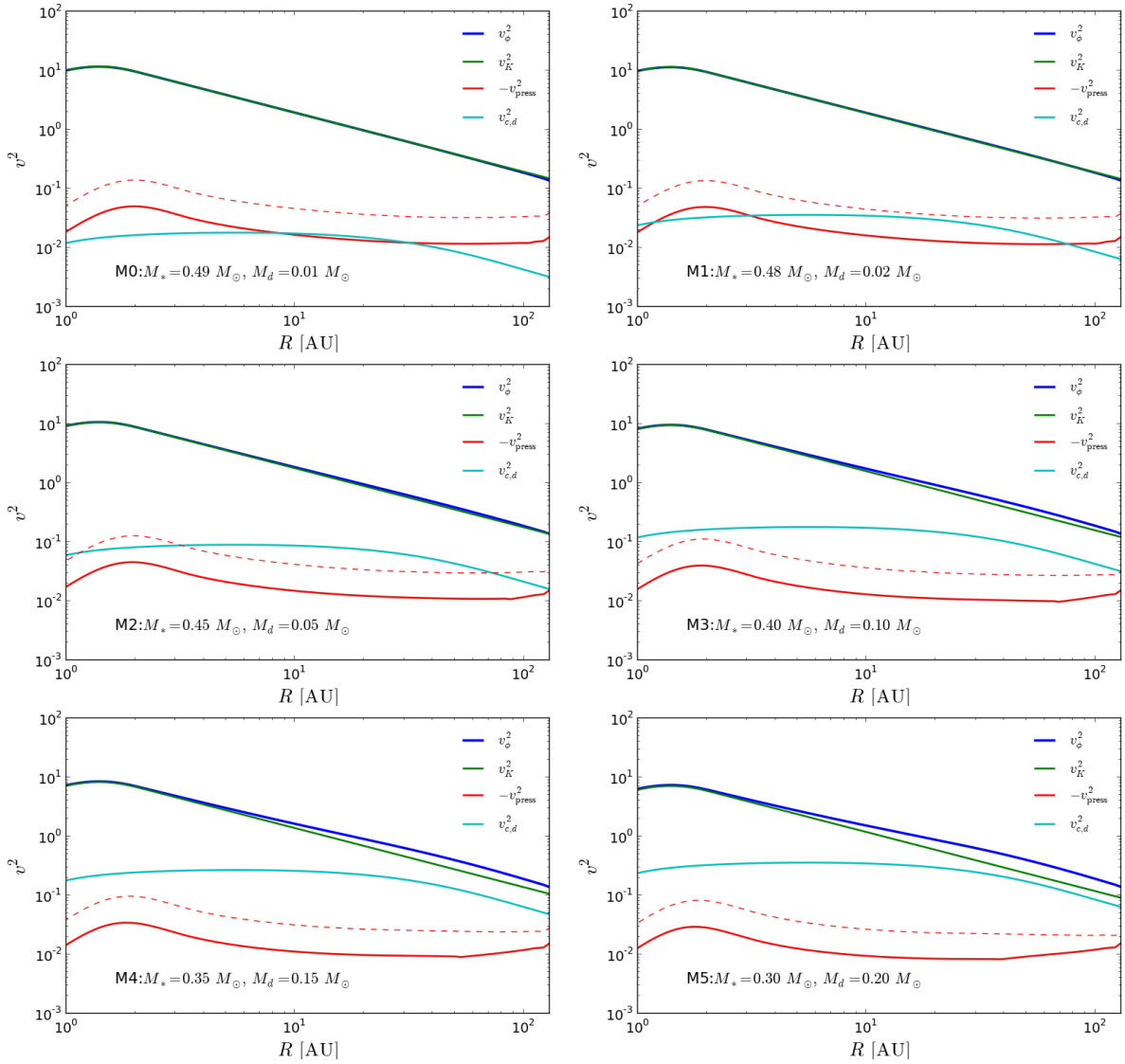


Figure 5.12.— Mid-plane velocity component profiles for the six disk profiles shown in Figure 5.1 in order of increasing disk mass. The panels include the total azimuthal velocity v_ϕ (blue curves), the Keplerian component v_K(green curves), the self-gravity component $v_{c,d}$ (cyan curves) and the pressure buffer term $v_{\rm press}$ (red curves). The solid red curves represent the thermal pressure contribution for a model with $h_{R_d} = 0.06$, while the dashed red curves represent models with $h_{R_d} = 0.1$.

cylindrical distance from the star and r is the spherical distance. Using the spherical distance, although unphysical, makes the calculation of the temperature much simpler. This is the method used by Shen et al. (2010).

5.3.4 Stellar accretion

Sink particles are a very important and necessary approximation in computational astrophysics. Sink particles allow for the removal of very high-density, gravitationally concentrated gas from the computational domain, while retaining the gravitational potential of the removed material by replacing it with a point-like particle. This procedure corresponds to effectively limiting the spatial resolution of the simulation, while adding a sub-resolution prescription that obeys some accretion formula, and that is designed to extend the calculation to scales below the spatially resolved regions of the flow.

Sink particles were introduced by Bate et al. (1995) for SPH. Since then, sink particle methods have been implemented for every major gas dynamics code for astrophysics, including ORION (Krumholz et al., 2004), GADGET (Springel et al., 2005a; Jappsen et al., 2005), FLASH (Federrath et al., 2010), ENZO (Wang et al., 2010b), RAMSES (Teyssier et al., 2011) and ATHENA (Gong & Ostriker, 2013).

The fact that this issue is revisited periodically shows that there is no universally correct way of implementing sink particles, and that a successful approximation will depend strongly on the numerical scheme for which it is implemented. Given the relative youth of moving-mesh methods in computational astrophysics, it is not clear yet which approach is the most appropriate for a code like AREPO. In principle, the

quasi-Langrangian nature of the code allows cells to be treated as SPH particles and, as such, they can be entirely "swallowed" by the sink if they satisfy some accretion criterion. In the primordial star formation simulations of Greif et al. (2011) cells within a certain accretion radius are swallowed if they are gravitationally bound to the sink. In the test-case cosmological simulations of Vogelsberger et al. (2012), cells are swallowed following the stochastic sampling of the black hole vicinity developed by Springel et al. (2005a) for the GADGET code. More recently, Vogelsberger et al. (2013) implemented a "draining" routine for cosmological simulations that makes direct use of the mesh-based discretization of AREPO. In this later case, black holes subtract mass from the cell within which they are located.

In this work we are interested in allowing the central star to accrete from its surrounding gas disks. It is desirable to allow the central gas to accrete any material that gets closer than some specified distance, while minimizing the effect of that this accretion might have on the rest of the computational domain. In standard accretion disk theory, the inner boundary condition determines the overall properties of the disk; since sink particles act as an innermost accretion boundary, how the flow is treated near this region can have a significant impact in the rest of the domain. For example, any aggressive removal of gas from the computational domain (for example mimicking a vacuum boundary condition around the star) can deplete the inner region of the disk. On the other hand, suppressing the removal of this material can cause, besides the previously mentioned small time steps, a pile up of gas that can cause reflections back into the rest of the disk.

Moving-mesh schemes can contain additional complications that are not present in either AMR or SPH. The quasi-Lagrangian nature of this method does not impose a

spatial resolution floor as AMR does. This implies that cells can spiral in arbitrarily close to the central star and pile-up at the bottom of the (softened potential), accumulating cells of nearly constant mass but ever decreasing volume, potentially generating a runaway growth of computing time. The first step we must take is to impose such a resolution floor and do not allow cells to become smaller than a certain reference volume within some distance from the star. By breaking the near mass constancy in this region, cells will increase in mass as numerical diffusion funnels gas onto the center of the disk. We then "shave off" the excess mass in these cells, and assign the corresponding fraction of mass and momentum to the central star.

5.4 More examples

5.4.1 Disk evolution with bulk motion

One of the main advantages of using a code like AREPO for circumstellar disk simulations is the minimization of the high Mach number problem. When solving the Euler equation in the rest frame of a moving cell, the local velocity of the flow is irrelevant for the outcome of the calculation. This can be a great advantage when simulating moving disks in binary or multiple stellar systems.

As an example we consider one of the examples of ICs shown in Figure 5.8 ($M_d = 0.05$, $M_* = 0.45$ and $h_{R_c} = 0.06$) and evolve it in isolation for ~ 1200 years. In addition, we take a copy of the same disk and give it a velocity of 6 km s^{-1} along the x-axis (i.e. perpendicular to the disk's symmetry axis). Within 1200 yrs, the boosted disk will cross the entire computational domain of length 2000 AU.

Figure 5.13 shows projections of the x-y plane for both the stationary disks and its moving copy at four different times. Although some axisymmetric structure develops with time, this is present in both the stationary and moving disks, showing that there is no noticeable effect on the structure of the disk (nor in the computing time) that can be attributable to bulk motion.

Similarly, Figure 5.14 shows the structure of the stationary and moving disks at the same points in time, but now projected onto the x-z plane. Again, the vertical structure of the disks shows little to no variability in time and no difference between the stationary and the moving cases.

We have also experimented with boosting the disk along the direction of its symmetry axis. Figure 5.15 shows the displacement of the disk along the x-direction, at a velocity of 6 km s^{-1} like before, but now the disk has been rotated. Again, the disk does not appear to be disturbed, and its thin-disk structure seems unaltered during the entire simulation. Note that in this case as the previous one, only the disk and its immediately surrounding mesh are moving with the central star; the background mesh, on the other hand, is initialized at rest. Presumably, at the extremely low densities of the background mesh, the effects of shocks and ram pressure take place when static cells meet stationary ones. However, given the low mass of those regions, the outcomes of such interactions are entirely negligible.

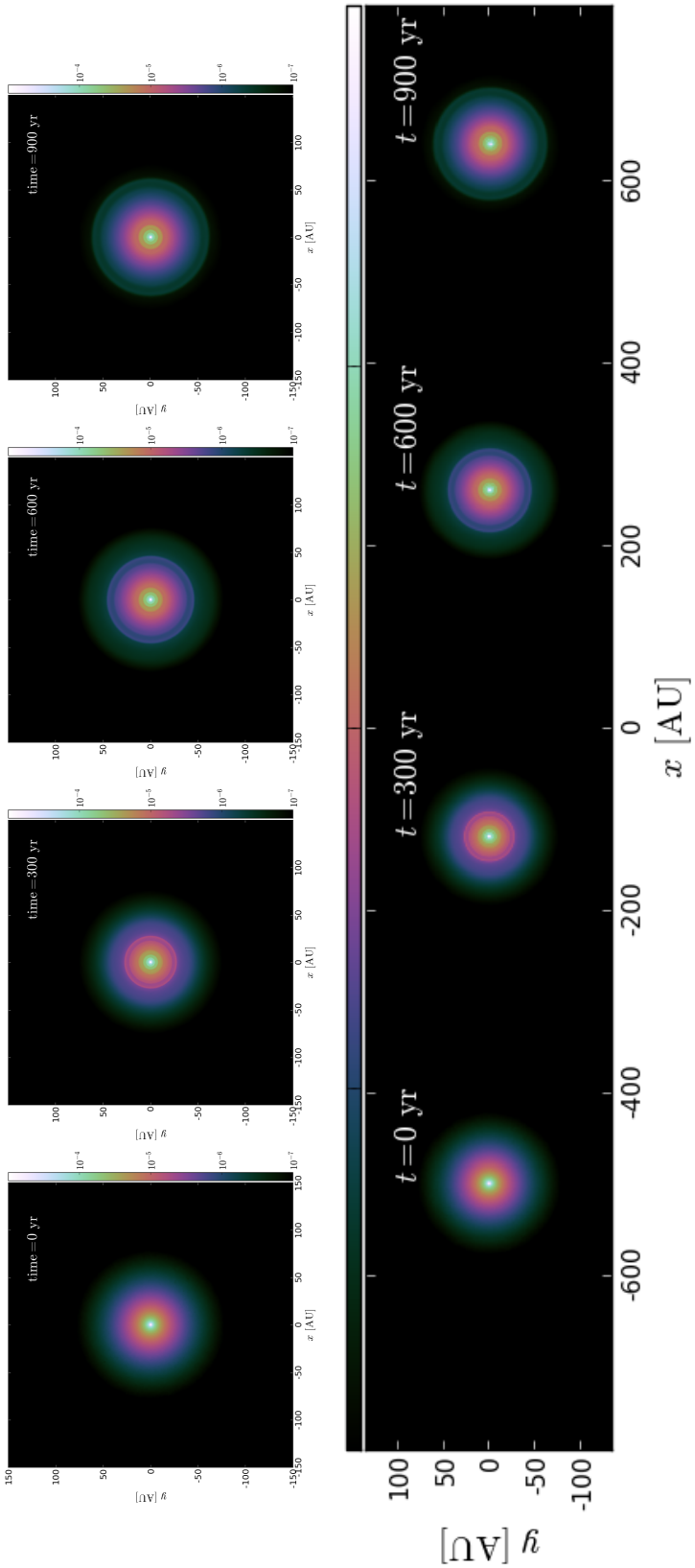


Figure 5.13.— Upper panels: face-on projection of a low-mass self-gravitating disk at four different times (left to right): 0, 300, 600 and 900 yrs. Lower panel: face-on projections of the same self-gravitating disk, now moving at a speed of 6 km s^{-1} along the x-direction, perpendicularly to the disk's axis of symmetry.

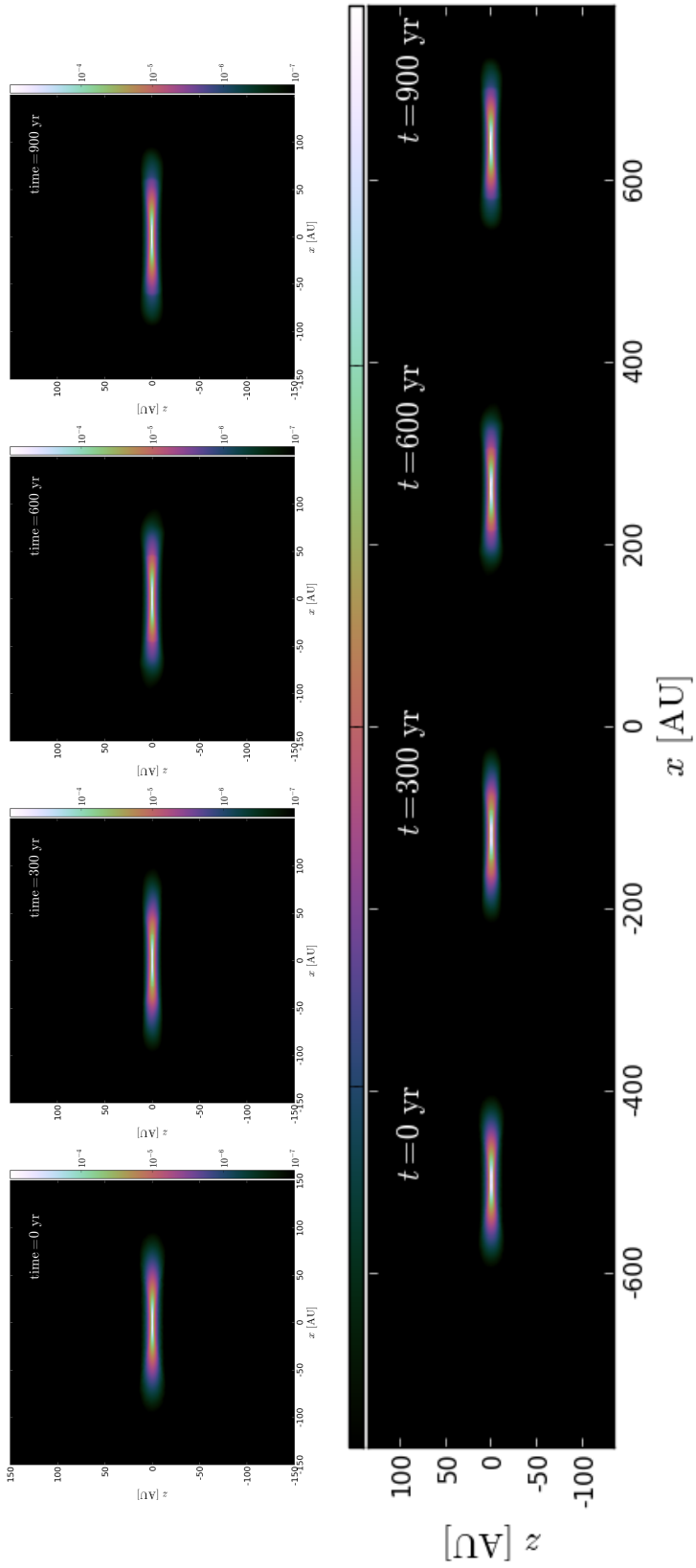


Figure 5.14.— Same systems as in Figure 5.13, but now for edge-on projections.

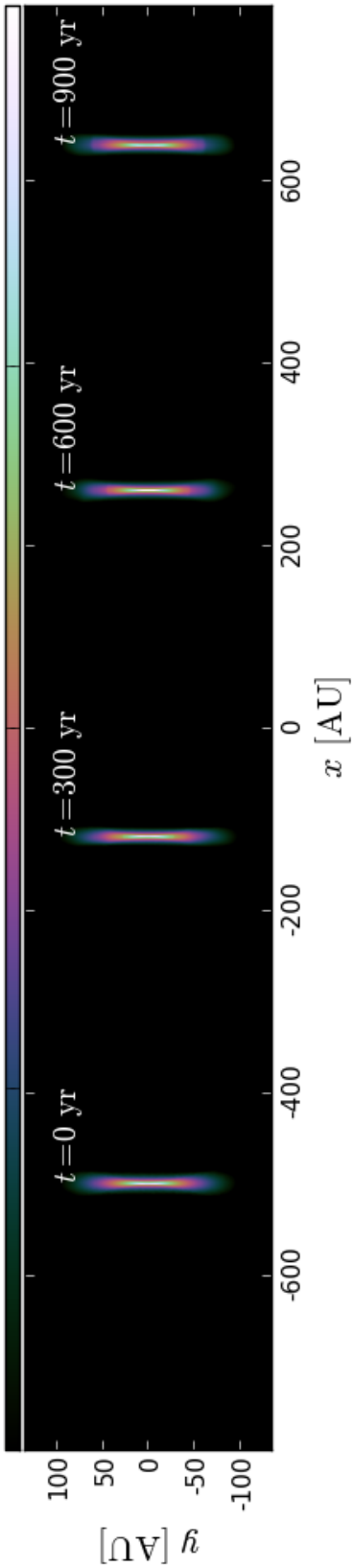


Figure 5.15.— Edge-on projections of a self-gravitating disk moving at a speed of 6 km s^{-1} along the x-direction, parallel to its axis of symmetry.

5.5 Appendix: Exact Rotation Curve for a Massive Lynden-Bell–Pringle Disk with $p = 1$

In the special case of $p = 1$, the Lynden-Bell–Pringle surface density profile takes the form

$$\Sigma(R) = \frac{M_d}{2\pi R_c^2} \left(\frac{R}{R_c}\right)^{-1} \exp\left[-\frac{R}{R_c}\right] \quad . \tag{5.14}$$

With this profile, the integrals in Equation 5.11 can be solved involving incomplete gamma functions:

$$\begin{aligned} v_{c,d}^2(R) = G\frac{M_d(<R)}{R} + \\ \frac{GM_d}{\pi R_c} \sum_{k=1}^{\infty} \alpha_k \Bigg[2k+1 \frac{R_c^{2k+1}}{R^{2k+1}} \gamma(2k+1, R/R_c) \\ - 2k \frac{R^{2k}}{R_c^{2k}} \, \Gamma(-2k, R/R_c)] \quad . \end{aligned}$$

However, a more efficient way of computing the exact rotation curve for the $p = 1$ case is to write the circular speed in terms of Bessel functions (Binney & Tremaine, 2008)

$$v_{c,d}^2(R) = 2\pi G R \int_0^{\infty} dk \, k J_1(kR) \int_0^{\infty} dR' R' \Sigma(R') J_0(kR') \quad . \tag{5.15}$$

For the case of $\Sigma(R)$ with $p = 1$, the innermost integral in Equation 5.15 can be computed with the aid of the identity

$$\int_0^{\infty} e^{-\alpha x} J_\nu(\beta x) dx = \frac{\beta^{-\nu} \left[\sqrt{\alpha^2 + \beta^2} - \alpha\right]^{\nu}}{\sqrt{\alpha^2 + \beta^2}}$$

(Gradshteyn & Ryzhik, 2000, §6.611), with $\alpha = 1/R_c$, $\beta = k$ and $\nu = 0$. This leaves

$$v_{c,d}^2(R) = GM_d \frac{R}{R_c} \int_0^{\infty} dk \, k \frac{J_1(kR)}{\sqrt{R_c^{-2} + k^2}} \quad .$$

which can be computed easily if we reformulate the integrand in terms of Meijer-G functions. Using

$$J_\nu(\gamma x) = G_{02}^{10}\left(\frac{\gamma^2 x^2}{4}\,\middle|\,\begin{matrix} - \\ \frac{1}{2}\nu, \quad -\frac{1}{2}\nu \end{matrix}\right)$$

(Gradshteyn & Ryzhik, 2000, §9.34), and

$$\int_0^\infty x^{\rho-1}(x+\beta)^{-\sigma} G_{pq}^{mn}\left(\alpha x\,\middle|\,\begin{matrix} a_1,...,a_p \\ b_1,...,b_p \end{matrix}\right) dx$$
$$= \frac{\beta^{\rho-\sigma}}{\Gamma(\sigma)} G_{p+1,q+1}^{m+1,n+1}\left(\alpha\beta\,\middle|\,\begin{matrix} 1-\rho, a_1,...,a_p \\ \sigma-\rho, b_1,...,b_p \end{matrix}\right)$$

(Gradshteyn & Ryzhik, 2000, §7.811), we obtain

$$v_{c,d}^2(R) = \frac{GM_d R}{2\sqrt{\pi}R_c^2} G_{13}^{21}\left(\frac{R^2}{4R_c^2}\,\middle|\,\begin{matrix} 0 \\ -\frac{1}{2}, \quad \frac{1}{2}, \quad -\frac{1}{2} \end{matrix}\right) \quad . \tag{5.16}$$

Note that

$$v_{c,d}^2(R) \xrightarrow[R\to 0]{} \frac{GM_d}{R_c} \quad ,$$

i.e., the rotation curve has a non-zero value at the origin. This is due to the divergent surface density at $R = 0$ in Equation 5.14, therefore, a softened surface density profiles is needed to reproduce a physically plausible self-gravitating rotation curve that increases from zero at the origin. This explains why the surface density softening cannot be independent from the gravitational softening of the central star, since it must be guaranteed that near $R = 0$ the rotation curve of the disk is entirely dominated by the point mass.

6

Orbital Evolution during Circumstellar Disk Encounters

6.1 Introduction

The complexity and diversity of exo-planetary system configurations revealed by *Kepler* data (e.g. Batalha et al., 2013) has exposed our limited theoretical understanding of the apparently very effi./chapter5/figures/cient process of planet formation throughout the Galaxy. One such exotic configuration that poses a challenge for the Solar nebula hypothesis of Safronov (1972) is that of planets orbiting both members of a binary system (circumbinary planetary systems, hereafter CPBs). At present, the occurrence rate of these systems is unconstrained, with 7 planets detected in 6 main-sequence systems Doyle et al. (2011); Welsh et al. (2012); Orosz et al. (2012). Planets have also been detected orbiting a single stellar member of a multiple system. Current detection statistics indicate a remarkably robust planet formation process in this configuration,

with statistical evidence showing an occurrence rate that is a significant fraction of that of isolated stars (Eggenberger et al., 2004; Bonavita & Desidera, 2007; Mugrauer & Neuhäuser, 2009). While these systems are more easily explained from a theoretical perspective , they nevertheless require an intimate connection between the star and planet formation processes.

When going from single star systems to binary systems, the planet formation process can be modified during at least three different stages (e.g. Zhou et al., 2012): (1) at the early circumstellar gas disk phase; (2) during the planetesimal formation stage; and (3) by altering the dynamics of already formed planetary systems at later stages. In this work, we focus on the first case by exploring the effects of primordial gas disk on the orbital evolution of their host protostars during close stellar encounters.

The morphology of circumstellar gas at early times can leave an imprint on the planet formation process by, for example, limiting the reservoir of gas available for formation, and by modifying the semi-major axis, inclination and eccentricity of the eventual planetary orbits. In binary systems, circumstellar material (dust and gas) can suffer large perturbations due to tidal truncation (e.g., Artymowicz & Lubow, 1994) or warping/bending (e.g., Larwood et al., 1996; Ogilvie & Dubus, 2001) or hastened disk dispersal (e.g., Alexander, 2012; Kraus & Ireland, 2012; Harris et al., 2012).

Understanding how planet formation proceeds in cluster environments is crucial for the development of a general theory that can connect the birth of stellar clusters in molecular clouds down to the small scale circumstellar environments that host planetary systems. Ultimately, to unveil the details of planet formation under these environmental conditions, we must understand the fate of the gas that remains in disks once the stars

have accreted most of their mass. (see Laughlin & Adams, 1998; Armitage, 2000; Bonnell et al., 2001; Adams et al., 2004, 2006; Fregeau et al., 2006; Fatuzzo & Adams, 2008; Gorti & Hollenbach, 2009; Gorti et al., 2009; Proszkow & Adams, 2009; Spurzem et al., 2009; Holden et al., 2011; Olczak et al., 2012; Pfalzner, 2013; Craig & Krumholz, 2013).

In the context of star clusters, star-star interactions are not limited to bound multiple systems, but may also include pairs (or higher order configurations) with zero or positive energy. Although hyperbolic (unbound) encounters between stars are essential for the dynamical relaxation of a cluster, their influence on spatial scales down to the circumstellar disk scales ($\lesssim 100$ AU) is expected to be a very small one; the probability of encounters with impact parameter b is proportional to b. Over the finite lifetime of (open) clusters, which is typically a few crossing times or ~ 1 to 3 Myr (Reggiani et al., 2011; Jeffries et al., 2011) with a small fraction reaching up to 10 Myr, interactions at distance of less than 200 AU are quite rare. As a consequence, planetary orbits within 30 AU of their host star should be essentially unaffected (Bonnell et al., 2001; Adams et al., 2006; Dukes & Krumholz, 2012), although some statistical evidence points toward the truncation of disks in more extreme clusters such as the Orion nebular cluster (ONC; see de Juan Ovelar et al., 2012).

Despite the rareness of events like direct star-disk and disk-disk interactions, such encounters have received significant attention in the literature, either focusing on their role in the tidal evolution of a binary and orbital capture (Clarke & Pringle, 1991a,b; Ostriker, 1994), or by studying the possible triggering of spiral arms, gravitational instability (GI) and fragmentation (Boffin et al., 1998; Lin et al., 1998; Pfalzner, 2003; Pfalzner et al., 2005; Pfalzner & Olczak, 2007; Forgan & Rice, 2009; Shen et al., 2010). There are several reasons to analyze these systems in detail. First, the evolution of

N-body simulations of clusters generated from semi-parametric models and not from *ab initio* simulations of star formation has been shown to be very sensitive to variables other than the mean stellar number density. For example, a dissolving cluster imposes a maximum timescale after which encounters cannot take place (Bonnell et al., 2001; Dukes & Krumholz, 2012; Craig & Krumholz, 2013). Also, sub-virial (Dukes & Krumholz, 2012) and structure-rich clusters (Craig & Krumholz, 2013) have produced significantly different results from idealized cluster models of Laughlin & Adams (1998); Adams & Laughlin (2001). Similarly, young cluster simulations *with gas* remain a challenging and expensive problem numerically (although see Hubber et al., 2013), and thus it is not truly known to what extent the presence of gas affects the encounter rate at times where the gas content of the cluster is a significant fraction of the total mass. Moreover, since stars form along filaments, they might begin their lives clumped relative to the average stellar density following dispersal (e.g. Offner et al., 2010). Secondly, as observations reach higher sensitivities and higher angular resolution, sample completeness makes rare examples accessible. Extreme environments like dense star clusters (e.g. the ONC), long-lived ones, or simply the much denser and permanently bound globular clusters are much more likely to host systems which have experienced close encounters. Third, small scale secular dynamics can alter the eccentricity of binaries and hierarchical multiples on timescales shorter than the lifetime of the disk but longer than the relaxation or evaporation time of the cluster. Circumstellar disks are observed in hierarchical multiple T Tauri stars (e.g. Andrews et al., 2010a), and processes like Kozai cycles (Kozai, 1962) could trigger close, eccentric encounters (e.g. Naoz et al., 2011; Lithwick & Naoz, 2011). Furthermore, even at late stages of stellar evolution in bound binaries, stellar mass loss could cause both the formation of "second generation" circumstellar disks (Perets &

Kenyon, 2013) and variation in the orbital elements that could excite the eccentricity (e.g., Perets & Kratter, 2012).

Even if star-disk and disk-disk encounters do take place in young clusters, they are nearly impossible to resolve adequately in the context of a full cluster hydrodynamical plus N-body simulation. Despite the increased sophistication of cluster models (Adams et al., 2006; Parker & Quanz, 2012; Bate, 2012; Craig & Krumholz, 2013),the dynamical range in timescales of N-body systems of stars with their respective planetary systems makes direct simulation of such systems prohibitive computationally. The limitations are not only owing to current hardware capabilities, but are also algorithmic in nature. For example, accurate N-body integrations of stellar clusters usually require direct force computations via, for example, Hermite integrators. However, these integrators can run into trouble after too many time steps, which would be the case if the spatial scales resolved went from the cluster scales ($\sim$ 1pc and crossing times of $\sim$ 1Myr) down to planetary orbit scales ($\sim$ 1pc and orbital perios of $\sim$ 1yr), implying a total integration time-to-time step ratio of at least 7 orders of magnitude, a dynamical range that can make the simulation computationally impractical. In addition, Hermite integrators are not symplectic, and thus suffer from cumulative errors (e.g. Aarseth, 2003) that can render the planetary orbits useless after hundreds of thousands to a few million orbits, which are the relevant timescales for cluster relaxation. Furthermore, accurate gravity solvers for clusters including gas as well as stars have only been recently begun to be explored (see the hybrid integrator of Hubber et al., 2013), in hopes to bridge the gap between gas-dominated star-formation simulations (e.g. Bate, 2012) and pure N-body simulations of young clusters. Therefore, a single full-blown direct N-body integration of a stellar cluster with planetary systems and circumstellar disks remains an extremely

challenging problem.

6.2 Numerical experiments on disk-disk interaction

6.2.1 Previous work

Numerical experiments on isolated configurations of star-disk and disk-disk interaction on spatial scales of ~ 100 AU enable the detailed study of the smallest spatial scales of clustered star formation at high resolution, accessing regions that are usually unresolved in self-consistent, *ab initio* simulations of star forming clouds with spatial scales of $\sim 10^5$ AU. Note that even the state-of-the-art star formation simulations of Bate (2012) –which produce hints of disks around protostars– have a mass resolution of $1.43 \times 10^{-5} M_{\odot}$, implying that a $0.01 M_{\odot}$ disk is composed of barely 700 resolution elements.

Studies of disk-disk collisions by direct numerical simulation date back to Lin et al. (1998) and Watkins et al. (1998a,b) (see also Boffin et al., 1998), when authors hypothesizes that condensation of material in tidally induced tails could produce a population of brown dwarfs. Later on, others authors have revisited the problem with both pure N-body approaches (Pfalzner, 2003; Pfalzner et al., 2005; Thies et al., 2005; Pfalzner & Olczak, 2007) and including with gas dynamics (Forgan & Rice, 2009; Sheppard & Trujillo, 2006; Shen et al., 2010; Thies et al., 2010). All the gas dynamical studies included self-gravity, although some of the N-body ones used test particles (e.g. Pfalzner et al., 2005), only focusing in the passive response of the disk to an external perturber. These studies have concentrated on the tidal generation of spiral arms (e.g.

Pfalzner, 2003); on disk fragmentation and formation of substellar mass objects (e.g. Lin et al., 1998; Sheppard & Trujillo, 2006; Shen et al., 2010; Thies et al., 2010) and disk truncation (e.g. Forgan & Rice, 2009). Those studies that include gas dynamics were all carried out using smoothed particle hydrodynamics (SPH; Lucy, 1977; Gingold & Monaghan, 1977; Monaghan, 1992; Springel, 2010b).

The fact that the numerical scheme of choice for disk-disk interaction has been almost exclusively SPH responds to a necessity of having a very adaptable scheme. On one hand, isolated circumstellar disks are axisymmetric to first order and, globally, nearly stationary structures. Therefore, they benefit from the use of structured grids in cylindrical coordinates, since such grid configurations favor low numerical diffusivity for azimuthal flow. On the other hand, as soon as this symmetry is broken (e.g., by combining two circumstellar disks moving at supersonic speeds toward each other), the benefit of structured grids becomes less clear. The meshless nature of SPH makes its performance independent of the geometry of the problem. In addition, its pseudo-Lagrangian adaptivity offers a robust and continuous resolution increase in regions of high density. These same reasons make this problem a tractable one for quasi-Lagrangian Eulerian codes like AREPO (Springel, 2010a), in which the control volumes evolve and move in a similar way SPH particles do, i.e., by following the local velocity field. By being locally a grid code, AREPO does not suffer from some of the numerical artifacts SPH is known to develop, such as clumping instabilities, suppression of hydrodynamic instabilities, artificial surface tension, zeroth order error terms, etc (see Vogelsberger et al., 2012; Sijacki et al., 2012; Bauer & Springel, 2012; Dehnen & Aly, 2012).

The study closest to our work is Shen et al. (2010), where the authors performed

fully self-gravitating simulations of gas disks encounters using SPH. Similarly to Lin et al. (1998), Shen et al. (2010) focused on the formation of self-gravitating objects (int the brown dwarf range) in tidal tails (reminiscent of the work of Barth, 1992) and in compression shocks during encounters of *very* massive disks (the values of star mass, disk mass and disk radius used area $M_* = 0.6 M_\odot$, $M_d = 0.4 M_\odot$ and $R_d = 250$ AU in Lin et al., 1998, and $M_* = 0.5 M_\odot$, $M_d = 0.6 M_\odot$ and $R_d \sim 1000$ AU in Shen et al., 2010). In this work, we model disks of more moderate –and more plausible– masses (10% of the mass of the star), focusing on the role of the tidal forces on the orbital evolution of the host stars, and exploring how small the impact parameter must be in order to cause a significant change to the original orbits.

6.2.2 Circumstellar disk models

Individual disk models are discussed in detail in Chapter 5. In this chapter, we use the same basic model, which consists of a self-consistent, self-gravitating solution for a disk with a surface density distribution that satisfies the Lynden-Bell–Pringle (Lynden-Bell & Pringle, 1974):

$$\Sigma(R) = (2-p)\frac{M_d}{2\pi R_c^2}\left(\frac{R}{R_c}\right)^{-p} \exp\left[-\left(\frac{R}{R_c}\right)^{2-p}\right] \quad . \tag{6.1}$$

where $M_d = 0.05\, M_\odot$, $p = 1$, $R_c = 20$ AU and stellar mass of $M_* = 0.45$ is used.

Our numerical scheme calculates full gas self-gravity and thus is able to capture disk fragmentation and object formation. However, the disk masses involved in our study make fragmentation much less likely than for the models of Shen et al. (2010). In order to avoid an excess of computer time dedicated to the high density regions undergoing gravitational collapse, we have used a temperature scaling high enough to minimize the

risks of disk fragmentation. None of the simulations performed for this work produce long-lived fragments. Still, the disk scale heights remain reasonable small ($H/R = 0.1$ at the disk characteristic radius) and the temperatures only reach a maximum of ~ 300 K when within 0.8 AU ($\approx$ the softening length of the stellar potential) and reach a floor temperature of 10 K at around 100 AU from the central star.

Besides computational convenience, there are two additional reason of why we have chosen to avoid fragment formation. First, our main focus is the orbital evolution of the stars following an encounter. Since the torque exerted on the stars by the disk gas depends on the mass distribution and not on the gas temperature, the results should not depend on whether or not a disk forms objects, unless of course that owing to low temperature (low Toomre Q) the fragmentation is so violent that the entire disk is turned into a few small objects such that the mass distribution is changed and, as a consequence, so is the torque. Second, realistic disk initial conditions for low mass stars are unlikely to spawn fragments over most of their lifetimes (Kratter et al., 2008, 2010).

Two relations are essential for determining the importance of tidal effects in disk-disk interactions. The first is the ratio between the disk size R_d and the pericenter distance between the two stars q. The second is the ratio between the disk's internal angular momentum $L_{\rm disk}$ and the binary system's orbital angular momentum $L_{\rm orb}$. If $q \gg R_d$ and $L_{\rm orb} \gg L_{\rm disk}$, tidal effects should be negligible and the stellar orbits should approximately evolve as those of point particles with mass equal to $M_* + M_d$. Although the disk's characteristic radius R_c (Equation 6.1) defines the scale at which the disk surface density profile transitions from power-law form to an exponential cutoff, it does not define a specific disk *size*. We define R_d as simply the radius that encloses 95% of the disk mass. For a surface density profile given by Equation 6.1, it can be shown that

the enclosed mass at radius R is

$$M_d(< R) = M_d \left\{ 1 - \exp\left[- \left(\frac{R}{R_c} \right)^{2-p} \right] \right\} \quad . \tag{6.2}$$

For $p = 1$, $R_d \approx 3R_c$, i.e. the disk size is approximately 60 AU in our models.

The disk internal angular momentum (measured respect to the center of mass of the star+disk system) $L_{\rm disk}$ can be computed analytically assuming that the azimuthal velocity field v_ϕ is well approximated by the Keplerian value v_K. For $p = 1$,

$$\begin{aligned} L_{\rm orb} &\approx 2\pi \int_0^\infty v_K(R) \Sigma(R) R^2 dR \\ &= \frac{\sqrt{\pi}}{2} M_d \sqrt{G M_* R_c} \end{aligned} \tag{6.3}$$

which is approximately 0.835 $M_\odot$ AU^2 yr^{-1}.. Computing $L_{\rm orb}$ from our 3D numerical modeles by directly summing over all cells gives a value of $L_{\rm orb} \approx 0.85$ $M_\odot$ AU^2 yr^{-1}.

Another quantity of interest is the disk's total energy $E_{\rm disk}$ (consisting of the disk's total kinetic energy, the total gravitational binding energy, and the total thermal energy). Since our simulations are initialized with the stars on parabolic orbits (see Section 6.2.3 below), the orbital energy of the system should be strictly zero in the limit of zero tidal effects, i.e., when the star+disk trajectories can be accurately represented by those of point particles. Therefore, any discrepancy between the total energy of the binary system at $t = 0$ and twice the value of $E_{\rm disk} \approx 0.24$ $M_\odot$ AU^2 yr^{-2} provides an indication of the tidal forces at startup and the validity of assigning point-mass trajectories to the disk-gas as initial conditions.

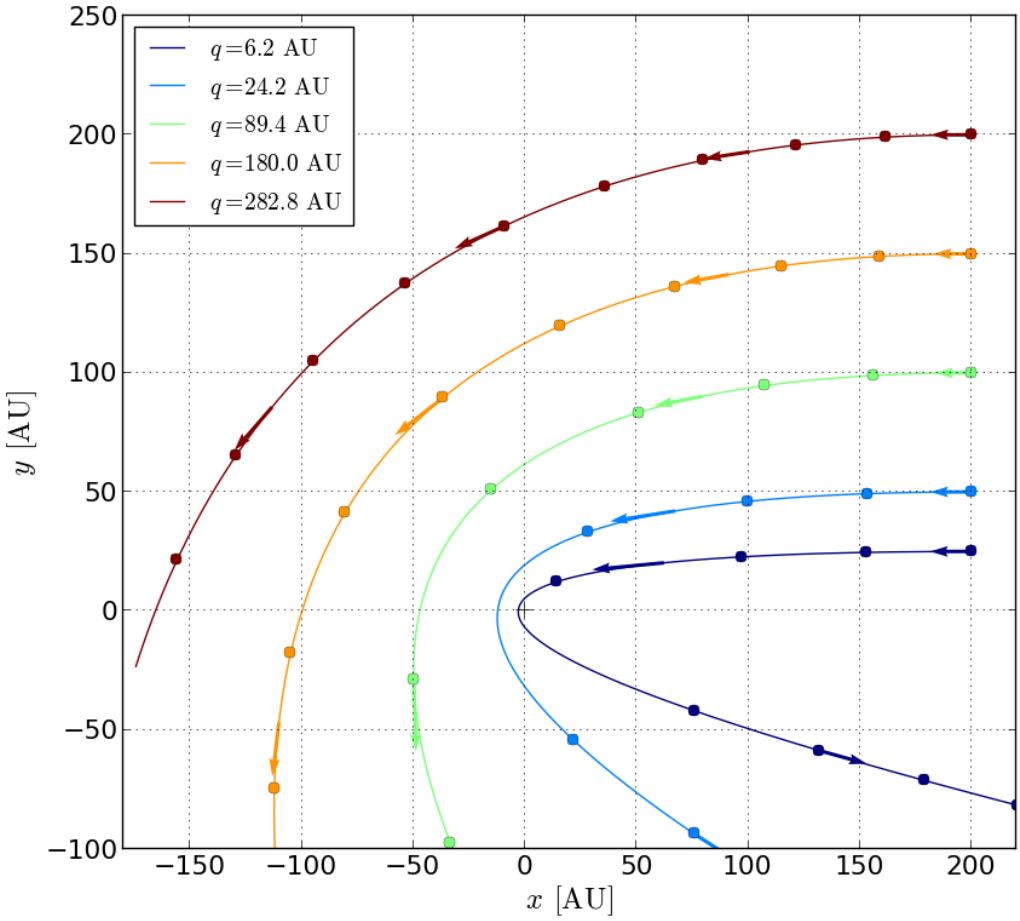


Figure 6.1.— Orbital configurations explored in this work (for clarity, only one component of the binary is shown). Five different parabolic orbits are setup with five different pericenter separations, which take values $q = 6.2$, 24.2, 89.4, 180.0 and 282.8 AU. Initial conditions star from the right of the figure at $x = -200$ AU for *all* modeled orbits (conversely, the binary component not shown here is started at $x = 200$ AU) with velocity along the x-axis. Orbital properties at $t = 0$ are calculated assuming each disk is a point particle of mass $0.5 M_{\odot}$. The magnitude of the velocity –chosen such that the orbit is parabolic for given an initial disk separation– ranges from ~ 1.7 km s^{-1} to ~ 2.1 km s^{-1}. The filled circles depict the locations for each trajectory in 200-year intervals.

Table 6.1

Simulation	q [AU]	θ_1 [°]	ϕ_1 [°]	θ_2 [°]	ϕ_2 [°]	$L_{\rm orb}$ [$M_\odot$ AU^2 yr^{-1}]	$L_{{\rm tot},z}$ [$M_\odot$ AU^2 yr^{-1}]	M_d/M_*	Disk-orbit alignment	Capture/fly-by
PARA1-1_M2	6.2	0	0	0	0	5.53	7.23	0.05/0.45	P-P	FB
PARA1-2_M2	”	0	0	45	0	”	6.98	”	P-P	C
PARA1-3_M2	”	0	0	90	0	”	6.38	”	P-P	C
PARA1-4_M2	”	0	0	135	0	”	5.78	”	P-R	C
PARA1-5_M2	”	0	0	180	0	”	5.53	”	P-R	C
PARA1-6_M2	”	180	0	180	0	”	3.83	”	R-R	C
PARA1-7_M2	”	180	0	135	0	”	4.08	”	R-R	C
PARA2-1_M2	24.2	0	0	0	0	10.9	12.6	0.05/0.45	P-P	FB
PARA2-2_M2	”	0	0	45	0	”	12.4	”	P-P	FB
PARA2-3_M2	”	0	0	90	0	”	11.8	”	P-P	FB
PARA2-4_M2	”	0	0	135	0	”	11.2	”	P-R	FB
PARA2-5_M2	”	0	0	180	0	”	10.9	”	P-R	C
PARA2-6_M2	”	180	0	180	0	”	9.23	”	R-R	C
PARA2-7_M2	”	180	0	135	0	”	9.48	”	R-R	C
PARA3-1_M2	89.4	0	0	0	0	21.0	22.7	0.05/0.45	P-P	FB
PARA3-2_M2	”	0	0	45	0	”	22.5	”	P-P	FB
PARA3-3_M2	”	0	0	90	0	”	21.9	”	P-P	FB
PARA3-4_M2	”	0	0	135	0	”	21.3	”	P-R	FB
PARA3-5_M2	”	0	0	180	0	”	21.0	”	P-R	FB
PARA3-6_M2	”	180	0	180	0	”	19.3	”	R-R	FB
PARA3-7_M2	”	180	0	135	0	”	19.6	”	R-R	FB
PARA4-1_M2	180.0	0	0	0	0	29.8	31.5	0.05/0.45	P-P	FB
PARA4-2_M2	”	0	0	45	0	”	31.3	”	P-P	FB
PARA4-3_M2	”	0	0	90	0	”	30.7	”	P-P	FB

Continued on Next Page...

Table 6.1 – Continued

Simulation	q [AU]	θ_1 [°]	ϕ_1 [°]	θ_2 [°]	ϕ_2 [°]	$L_{\rm orb}$ [$M_\odot$ AU^2 yr^{-1}]	$L_{{\rm tot},z}$ [$M_\odot$ AU^2 yr^{-1}]	M_d/M_*	Disk-orbit alignment	Capture/fly-by
PARA4-4_M2	”	0	0	135	0	”	30.1	”	P-R	FB
PARA4-5_M2	”	0	0	180	0	”	29.8	”	P-R	FB
PARA4-6_M2	”	180	0	180	0	”	28.1	”	R-R	FB
PARA4-7_M2	”	180	0	135	0	”	28.4	”	R-R	FB
PARA5-1_M2	282.8	0	0	0	0	37.4	39.1	0.05/0.45	P-P	FB
PARA5-2_M2	”	0	0	45	0	”	38.8	”	P-P	FB
PARA5-3_M2	”	0	0	90	0	”	38.2	”	P-P	FB
PARA5-4_M2	”	0	0	135	0	”	37.6	”	P-R	FB
PARA5-5_M2	”	0	0	180	0	”	37.4	”	P-R	FB
PARA5-6_M2	”	180	0	180	0	”	35.7	”	R-R	FB
PARA5-7_M2	”	180	0	135	0	”	35.9	”	R-R	FB

Table 6.1:: Orbital parameters for parabolic encounters with different pericenter separations.

6.2.3 Orbital configuration

Two identical copies of the fiducial disk model presented in Section 6.2.2 are used to setup a parabolic encounter. Consequently, the orbital energy of the binary is $E_{\rm orb} = 0$. All orbits live on the x-y plane and are initialized with the same x-coordinate. The initial velocity is directed along the x-axis (the disks directed toward each other). Therefore, the only free parameter is the pericenter separation q. Figure 6.1 shows the initial trajectories of one of the disks for five different values of q: 6.2, 24.2, 89.4, 180.0 and 282.8 AU. We call these different parabolic orbit configurations `'PARA1'`, `'PARA2'`, `'PARA3'`, `'PARA4'` and `'PARA5'` respectively. In addition, we vary the orientation of the disks respect to the orbital angular momentum vector (angles θ_1 and θ_2). We vary the disk orientations in seven different configurations, which are labeled accordingly by appending a number to the orbital label, e.g., `'PARA1-1'`, `'PARA1-2'`, etc. Table 6.1 shows the the main set of simulations and their respective orbital and orientation parameters. Each orbital configuration (set by the value of q) contains seven variants, which correspond to different combinations of the angles θ_1 and θ_2 (same notation as Shen et al., 2010). The azimuthal orientation of the disks (angles ϕ_1 and ϕ_2) is not changed.

The orbital angular momentum in the two body problem is $L_{\rm orb} = m\sqrt{\mu q(1+e)}$ where $m = M_1 M_2/(M_1 + M_2) = 0.25 M_\odot$ is the reduced mass and $\mu = G M_{\rm tot}$ where the total mass $M_{\rm tot} = 1 M_\odot$. The seventh column in Table 6.1 shows the orbital angular momentum of each simulation according to the chosen value of q. The eighth column shows the expected value of the z-component of the *total* angular momentum $L_{{\rm tot},z}$,

taking into account the contribution from the disk internal angular momentum, i.e.,

$$L_z = L_{\rm orb} + L_{\rm disk,1}\cos\theta_1 + + L_{\rm disk,2}\cos\theta_2 \quad , \tag{6.4}$$

where $L_{\rm disk,1} = L_{\rm disk,2} \approx 0.85$ (Section 6.2.2). A comparison between $L_{\rm orb}$ and $L_{{\rm tot},z}$ shows the whether the total angular momentum can be changed significantly by simply changing the orientation of the disks. From these quantities, we can estimate that the simulation subsets `'PARA1'` and `'PARA2'` should show a greater degree redistribution of angular momentum between the gas and the stars (and produce capture) as well as a significant dependence of the simulation outcome on the orientation of the disks. Our initial conditions satisfy Equation 6.4 within less than a few percent, indicating that the superposition of two stationary, isolated disk models into a self-interacting binary system is reasonably adequate at the values of the initial separation D that we have chosen. We have noticed that the angular momentum error seeded on startup is slightly larger for the larger pericenter simulations. Although the disks in these simulations were started far apart (Figure 6.1) precisely to avoid these problems, it is worth pointing out that the angular momentum of the system grows faster with q than with D. For example, the ratio in D for the orbital configurations `'PARA5'` ($q = 282.8$) and `'PARA1'` ($q = 6.2$) is ~ 1.4, while the ratio in $L_{\rm orb}$ for the same configurations is ~ 6.8. As a consequence, the errors in setting up the orbit are fractionally larger, albeit still small, for our wider orbits. Although the impulsive forces implied by this fractional error are still negligible for the dynamics of the encounter, we foresee some complications arising from not setting up a configuration that is not "asymptotic enough".

Similarly, the consistency between the measured energy at the zeroth snapshot and the expected energy derived from the superposition of two isolated disk models provides

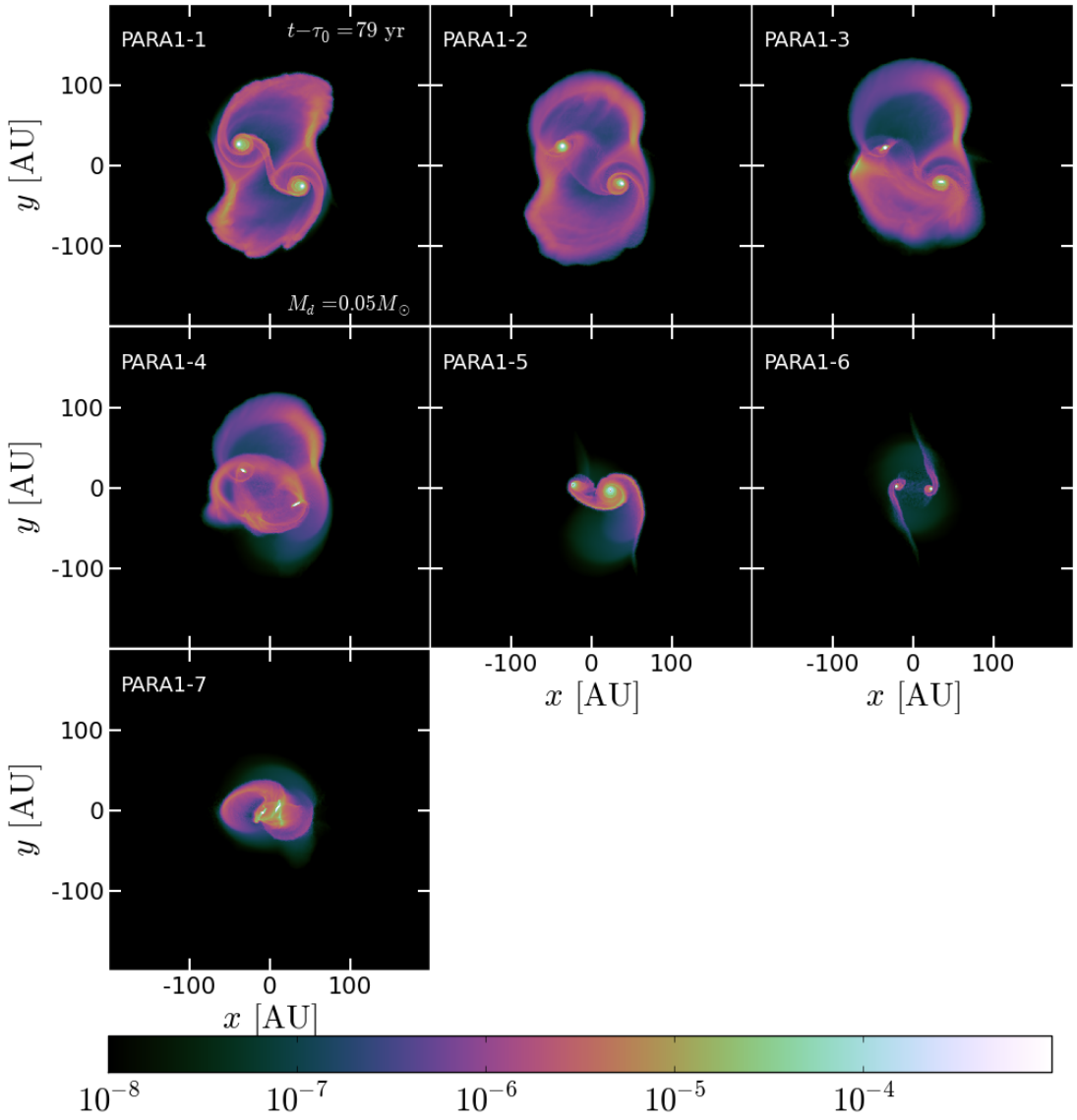


Figure 6.2.— Simulation output for the orbital set `'PARA1'` ($q = 6.2$)a short time after pericenter passage ($t - \tau_0 = 79$ yr), which corresponds to a simulation time of 700 yr. Six out of these seven simulations show orbital capture before the end of the integration (5000 years), meaning that the stars came back for at least one more pericenter passage (see text). Each frame shows the projected density in units of $M_\odot$ AU^{-2} (the conversion factor to g cm^{-2} is $\approx 8.9 \times 10^6$). All images are generated by integrating the three-dimensional density field along one direction following the full Voronoi mesh.

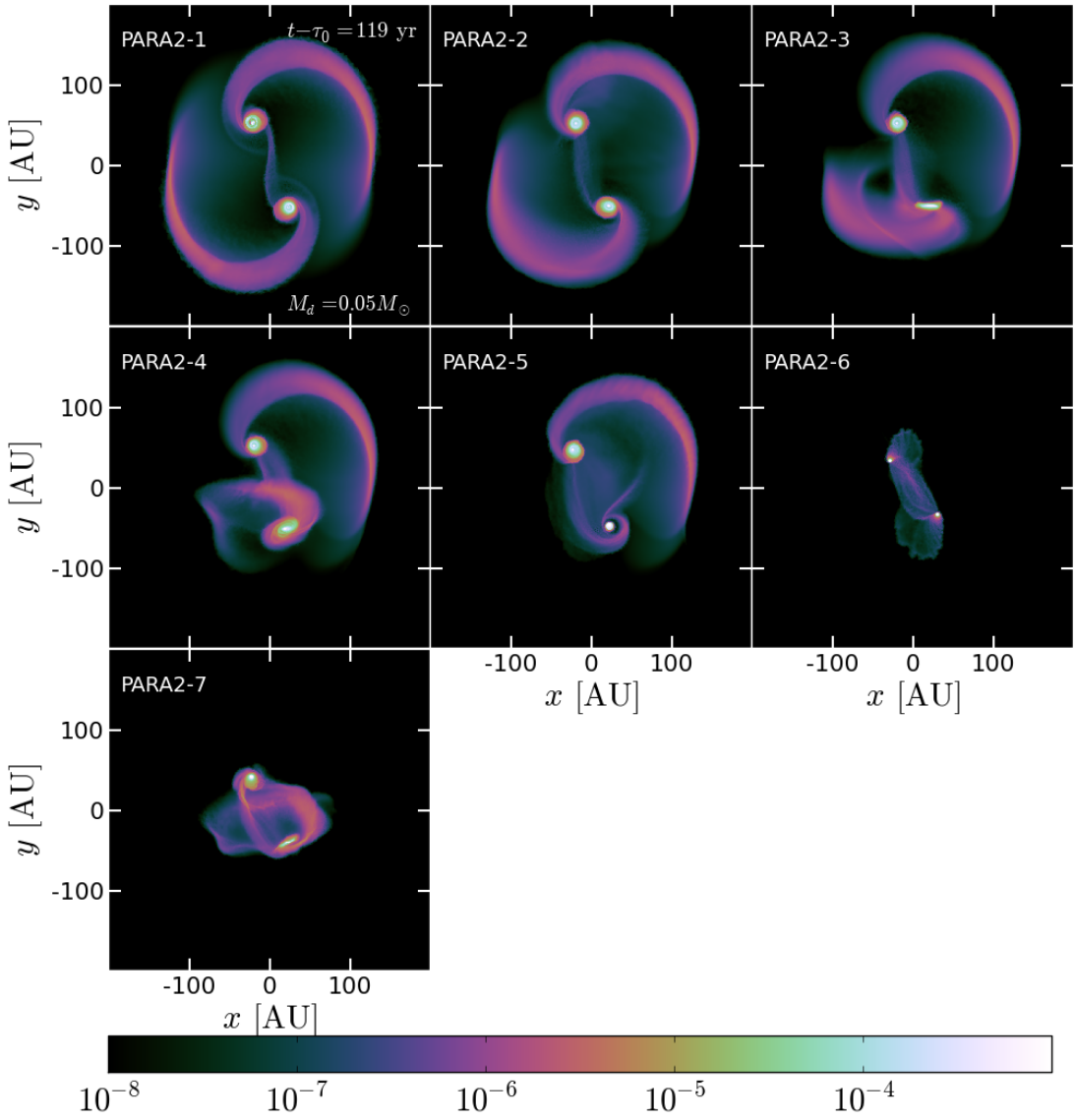


Figure 6.3.— Simulation output for the orbital set `'PARA2'` a short time after pericenter passage ($t-\tau_0 = 119$ yr), which corresponds to a simulation time of 800 yr. See description of Figure 6.2.

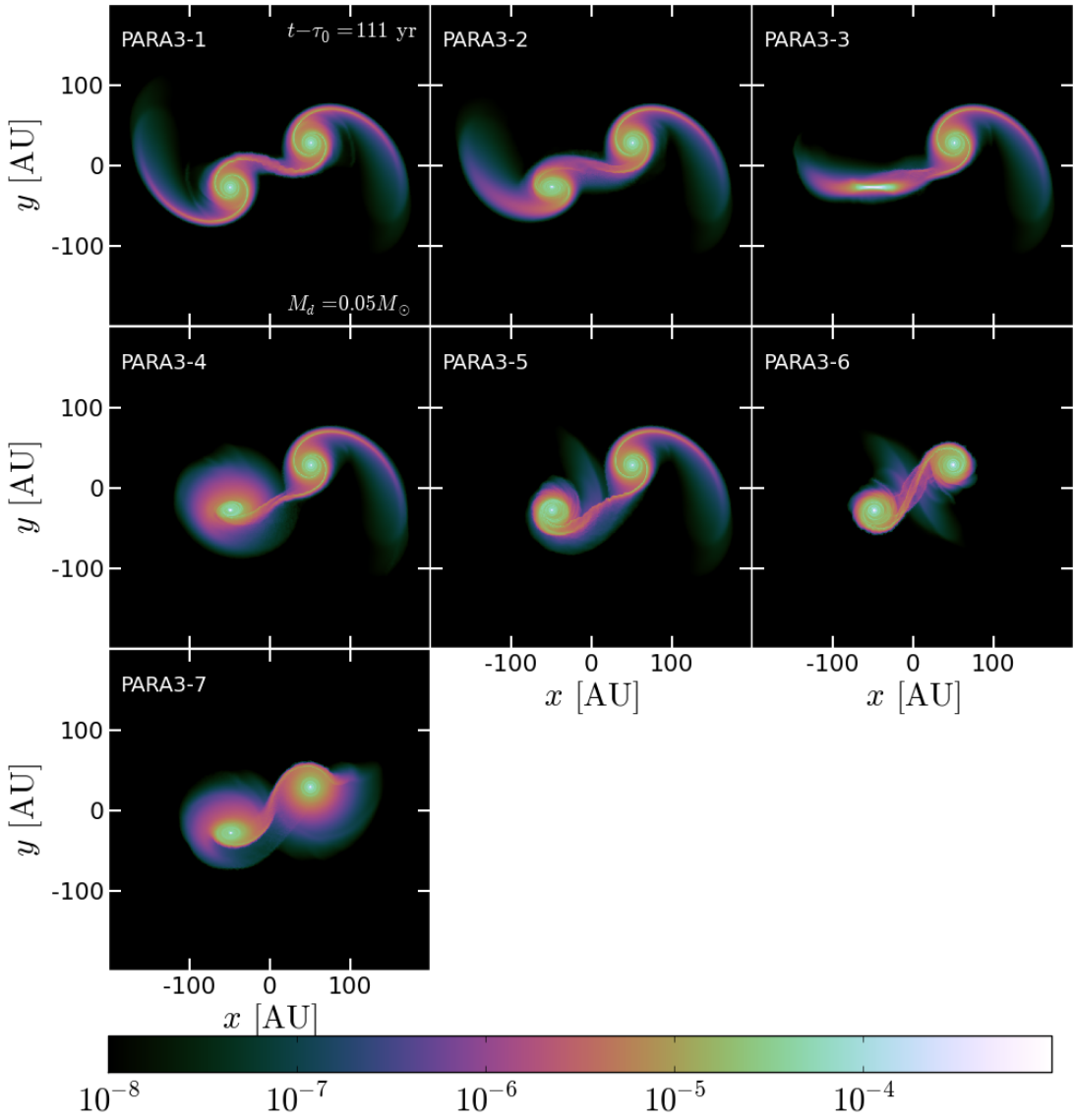


Figure 6.4.— Simulation output for the orbital set `'PARA3'` a short time after pericenter passage ($t - \tau_0 = 111$ yr), which corresponds to a simulation time of 1000 yr. See description of Figure 6.2.

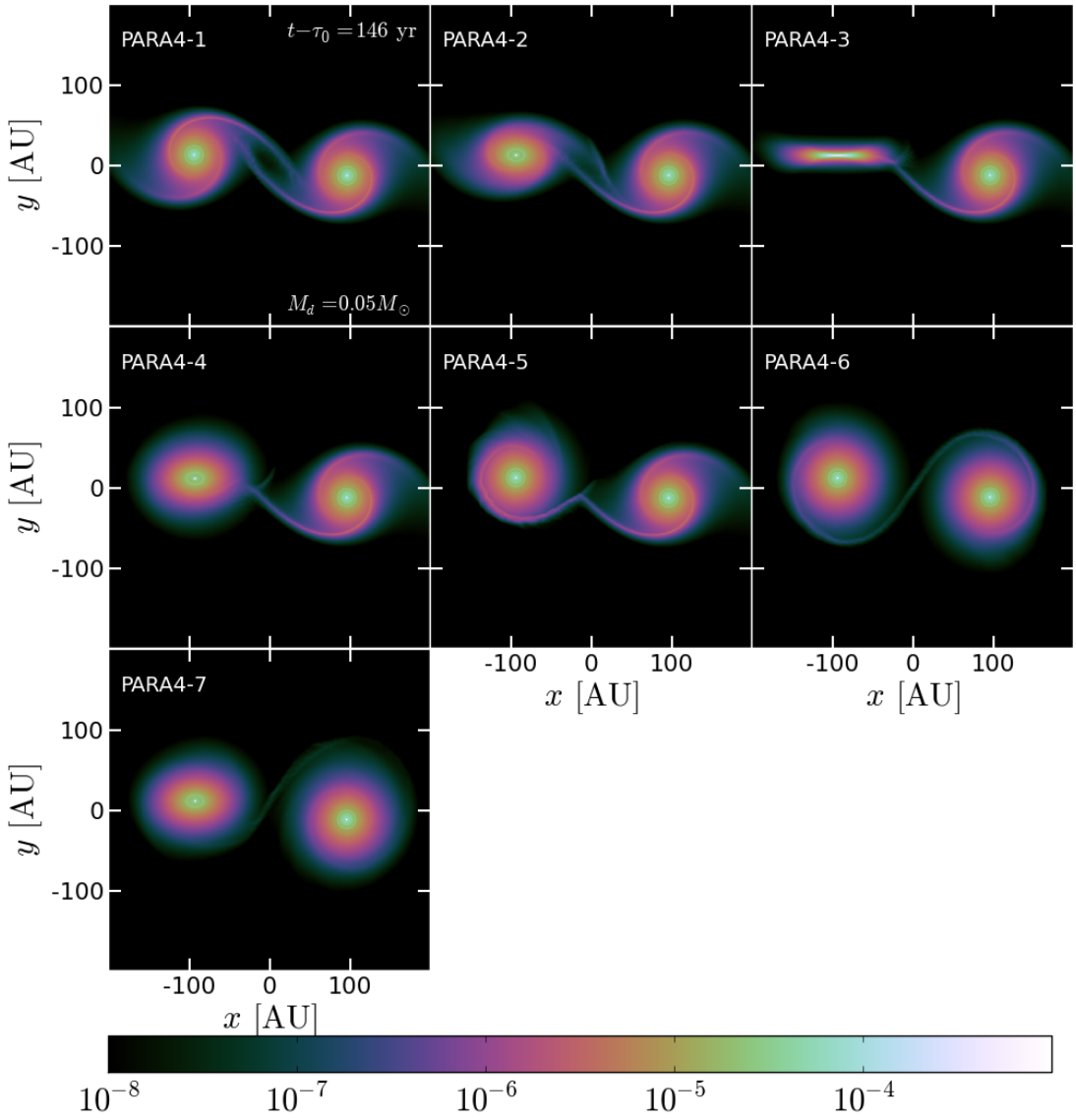


Figure 6.5.— Simulation output for the orbital set `'PARA4'` a short time after pericenter passage ($t - \tau_0 = 146$ yr), which corresponds to a simulation time of 1300 yr. See description of Figure 6.2.

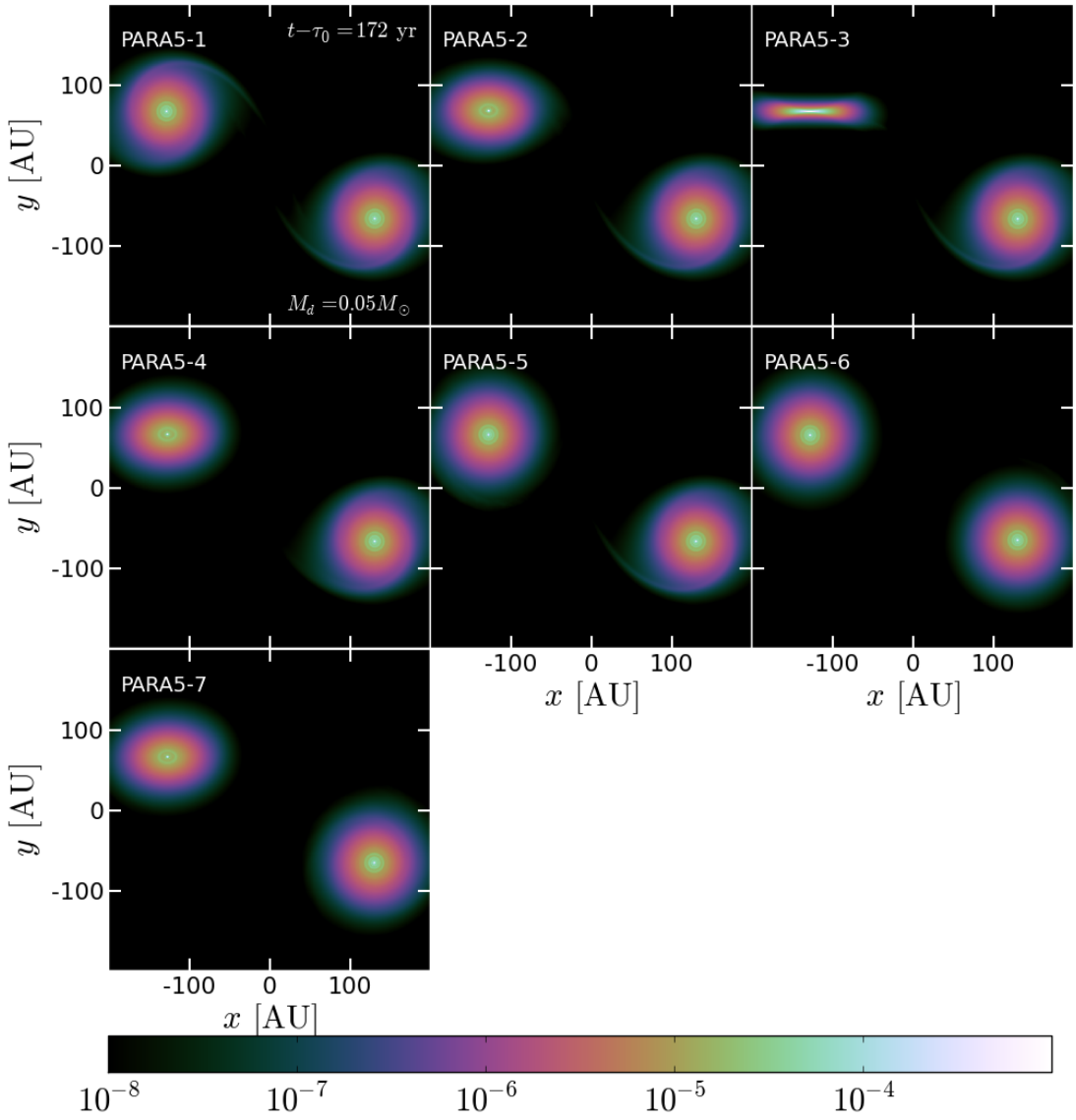


Figure 6.6.— Simulation output for the orbital set `'PARA5'` a short time after pericenter passage ($t - \tau_0 = 172$ yr), which corresponds to a simulation time of 1600 yr. See description of Figure 6.2.

another measure of whether the initial conditions are quiet, or if the disks are so close that the tidal forces are imparted impulsively at the beginning of the simulation. The total energy of the system is $E_{\rm tot} = 2E_{\rm disk} + K_{\rm orb} + V_{\rm orb}$, but as we have mentioned above, $K_{\rm orb} + V_{\rm orb} = E_{\rm orb} \approx 0$. Therefore, the total energy of the system, *regardless* of the pericenter distance, should be just twice the energy of the individual disk model: $E_{\rm tot} \approx -0.48$. Just as with $L_{\rm tot}$, we find deviations of the order of $1 - 2\%$, indicating that some tidal forces are acting on the disks at $t = 0$.

6.3 Results

6.3.1 Encounter morphology

Figures 6.2-6.6 show projected density images of all 35 simulations listed in Table 6.1. Each of these figures shows a simulation subset `'PARA1'` to `'PARA5'` (Table 6.1) at some time after pericenter.

Simulation sets `'PARA1'` and `'PARA2'` have pericenter distances of $q = 6.2$ and 24.2 respectively, and therefore we expect the greatest disruption to the gas disks in these simulation sets due to tidal effects but also due to direct shock-induced truncation. The disk models have characteristic radii of $R_c = 20$ AU and outer radii of $R_d = 60$ AU (Section 6.2.2), and thus the disks are expected to collide directly (i.e., $q/2 < R_d$) for configurations `'PARA1'`, `'PARA2'` and `'PARA3'`. However, for `'PARA3'`, the enclosed mass at $R = q/2 = 44.7$ is 89% of the disk total mass, and thus, despite the evident disk truncation seen in Figure 6.4 (note that the projected density range spans nearly 5 orders of magnitude), this encounter should have little effect on the stellar orbits.

Indeed, two different regimes can be clearly distinguished if the orbits of the stars are considered. Figures 6.2 -6.3 show significant disruption of the gas distribution, but also to the orbital evolution of the stars. Figure 6.4 shows some truncation of the disks and strong tidal features, but the disk centers (approximately the position of the stars) shows no distinguishable variation from frame to frame, i.e., the orientation of the disks bears little importance, an indication that the response of the disk orbits depends weakly on the extended mass distribution and that it can be reasonable approximated to those of two point masses. Figures 6.5 and 6.6 show little to no modification in the disk surface density besides the excitation of $m = 2$ spiral arms, which are characteristic of tidal encounters (e.g. Binney & Tremaine, 2008; D'Onghia et al., 2010). Evidently, the steepness of the tidal force with distance can explain the rapid change in output with q (Ostriker, 1994 calculations show an exponential dependence of the energy change in $(q/R_d)^{3/2}$).

Another indication of the rapidly decreasing influence of the disk is the ubiquitousness of some tidal features. Recalling from Table 6.1 that simulations `'PARA_-1'- 'PARA_-5'` differ only in the orientation of disk #2 while keeping the symmetry axis of the disk #1 aligned with the z-axis, we can see the same tidal features in disk #1 in all first five panels of Figure 6.4-Figure 6.6 and to a significant extent in Figure 6.3 (these morphological similarities are harder to find in Figure 6.2), regardless of the orientation and shape of disk #2, an indication that to first order the tidal features depend on the monopole component of the companion's potential (strongly dominated by the central star), even as significant mass stripping has taken place due to the physical collision of the disks.

6.3.2 Stellar orbits

Figure 6.7 shows the inter-star separation for the most disruptive set of orbits (`'PARA1'`). At pericenter passage, all orbits suffer a significant energy loss that shrinks the semi-major axis compared to the initial parabolic trajectory. Before pericenter (i.e., t <621 yrs) all runs follow the analytic trajectory closely, and reach pericenter at the same time. After pericenter, the orbital evolution varies dramatically among these configurations. In six of the runs, there are at least three additional close pericenter passages and over ~ 50 additional ones in the most disruptive configuration. Based on these additional separation minima we categorize these systems as "captured." The seventh configuration (`'PARA1-1'`) hints that the stellar separation has reached apocenter at the end of the simulation, and that it should go back for a second passage at around $t = 7000$ yr. Consequently, all simulations in the set `'PARA1'` show sufficient energy loss to be considered bound after first passage.

The different outcomes of the `'PARA1'` simulations is determined by the relative orientation of the disks. Each curve in Figure 6.7 is labeled according to a normalized z-spin value $S_z \equiv \cos\theta_1 + \cos\theta_2$. Prograde-prograde encounters like `'PARA1-1'`, `'PARA1-2'` and `'PARA1-3'` contain a larger amount of angular momentum than the prograde-retrograde and the retrograde-retrograde encounters. Configurations that include one or two disks in retrograde orientation will result in different torques on to the stars since the initial response of the gas to first passage will be different. Retrograde orientations do not contain orbital resonances (e.g. Toomre & Toomre, 1972; D'Onghia et al., 2010), and extended spiral arms are not formed as a result.

Interestingly, the orbital decay is almost entirely determined by the energy loss at

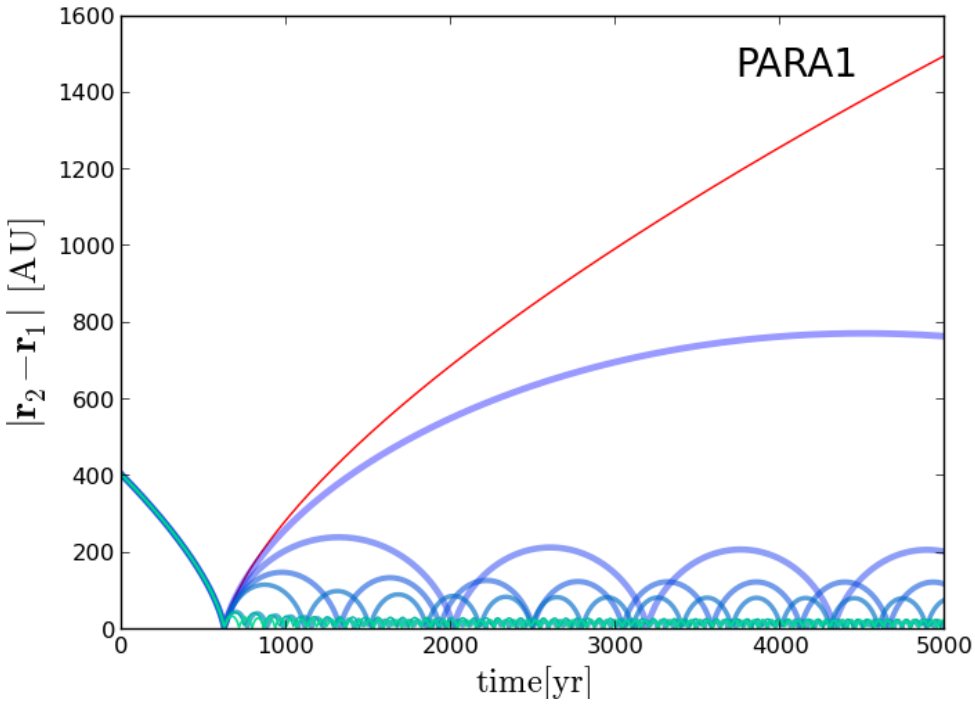


Figure 6.7.— Time evolution of the stellar separation for all simulations in the `'PARA1'` orbital configuration. Lines are labeled according to "normalized total disk spin" in the z-direction, simply defined as $S_z = \cos\theta_1 + \cos\theta_2$, and change in color and thickness according to the value of S_z. A total spin value of 2 implies that both disks are have angular momentum exactly aligned with the orbital angular momentum vector. A value of -2 implies that both spins are antiparallel with the orbital angular momentum vector. The thin red line represents the exact solution of the initial two-body problem assuming the disks are point masses. The simulation results show the red line is followed very closely by the stars until pericenter passage.

first passage. Although the amplitude of the separation curve is observed to decay slowly in time, most of the dissipation happens at once when the disks first meet. This is not surprising given that configuration `'PARA1'` has a pericenter distance ($q = 6.2$) that is significantly smaller than the sum of the disk radii ($2 \times R_d = 120$ AU). Thus, the first encounter violently truncated the disk on very short timescales, potentially reducing the mass of the disk by a factor of ~ 4 (from Equation 6.2, $M_d(< 6.2\,\mathrm{AU}) \approx M_d/4$), after which the tidal interaction goes back into a linear regime and the orbit evolves more slowly.

Figure 6.8 shows the evolution of stellar separation for the rest of the simulations in our study, grouped by orbital configuration. The likelihood of capture decreases very rapidly with pericenter separation. Only three out of seven simulations in the `'PARA2'` show additional pericenter passages (although all of them show substantial orbital energy loss). The runs in set `'PARA3'` show a much weaker effect; although, as before, the change in separation increases when S_z is decreased. The stars should not be expected to interact again for another few 10,000 yrs, and after reaching separations of a few to several thousand AU, meaning that these systems are not true "binaries". In the case of `'PARA4'` and `'PARA5'`, the interaction appears extremely weak, since stellar separations remain on their original parabolic trajectories with variations of the order of 1% toward the end of the simulation. These variations are attributable to other effects besides tidal interactions (e.g., numerical accuracy, no strict conservation of angular momentum, torque onto the stars by gas accretion, etc). Although the tidal response of the disk is clear in these last two examples, the long term effect on the orbits of the stars is negligible.

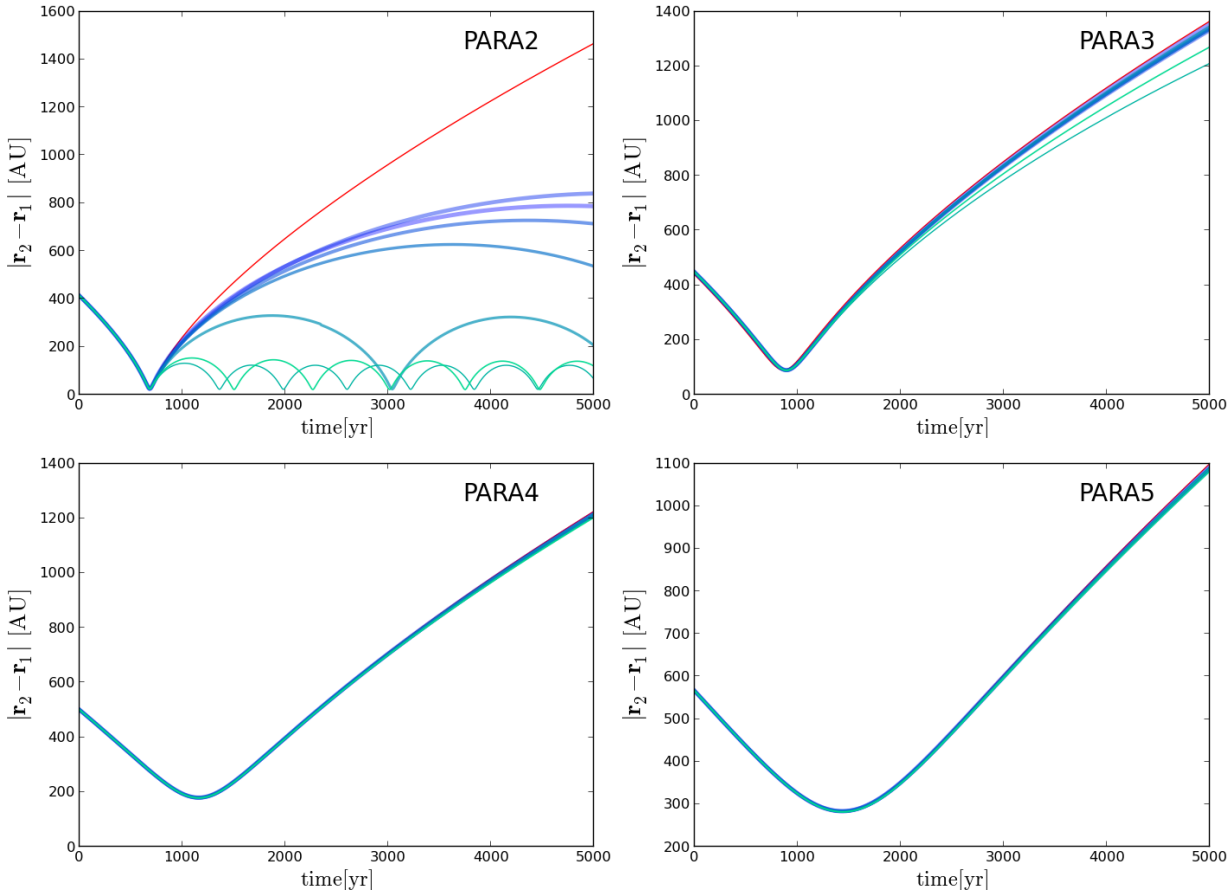


Figure 6.8.— Same as Figure 6.7, but now for the 'PARA2', 'PARA3', 'PARA4' and 'PARA5' configurations. As before, the color of the curves is chosen according to the combined disk spin in each configuration

6.3.3 Orbital evolution

Although the stellar separation is informative, it does not necessarily describe completely the orbital evolution of these star+disk systems. In order to analyze the orbital evolution of each star+disk system, we need to assign them meaningful osculating elements. However, identifying what gas belongs to a disk and which is simply is surrounding material is not a trivial task. Although group finding algorithms abound in the literature, these rely on heavily clustered distributions in space (this is the basis for halo-finding algorithms in cosmology; e.g., Davis et al., 1985). Although spatial density-based clustering algorithms might have little trouble identifying disks in configurations like that of Figure 6.6, disordered gas distributions like that of Figure 6.2 might present a great challenge for automized searches. For simplicity, we will consider to be disk material any cells lying within each star Roche lobe (in this equal-mass example, delimited roughly by the midpoint between the two stars). Then, we proceed to identify all the mass within the chosen region, calculating its center of mass and the center of mass velocity. With these quantities, we define a classical two-body problem and calculate the orbital elements for each snapshot. Since over short timescales the dynamics should be dominated by the two stars (they contain 90% of the mass of the system), we expect this approximation to be a good first order indicator of the orbital evolution of the system.

Figure 6.9 shows the orbital elements calculated in the way described above for all 35 simulations. The pericenter time series show a markedly different behavior between simulation sets `'PARA1'` and `'PARA2'` with respect to sets `'PARA3'`, `'PARA4'` and `'PARA5'`. The former group shows substantial changes in q after pericenter (up to 50% in the case of `'PARA1'`), while the latter group shows a variability that could

be consistent with a random walk behavior. Smoother over timescales of ~ 1000, sets `'PARA3'`, `'PARA4'` and `'PARA5'` show little to no consistent trend in q as a function of time. The total Δq ranges from $1\% - 2\%$ in the case of `'PARA3'`, to $\sim 0.1\%$ in the case of `'PARA5'`. The seemingly stochastic behavior of q in this regimes leads us to conclude that the evolution is dominated by numerical noise or stochastic accretion due to our sink particle scheme and that the "true" small variability of q is buried (`'PARA4'` and `'PARA5'`) or partially buried (`'PARA3'`) under said noise.

The evolution of eccentricity shows a significant deviation from unity for sets `'PARA1'` (up to 30%), `'PARA=2'` (up to 30%) and `'PARA3'` (up to 6%). Even `'PARA4'` and `'PARA5'`, although again with a noticeable component of stochasticity, show a clear overall trend of decreasing eccentricity that flattens out toward the end of the simulation. The fact that the eccentricity reaches a finite value toward the end proves that, despite the evident noise contamination, this loss of energy is real; furthermore, `'PARA5'` flattens out later than `'PARA4'`, consistently with the fact that the pericenter timescale of `'PARA5'` is longer and therefore the tidal interaction is expected to be spread over a longer period of time.

An interesting outcome of the pericenter evolution of `'PARA1'` is that q grows after pericenter for positive values of S_z but decreases for $S_z \leq 0$. Therefore, although Figure 6.7 already hints at loss of orbital energy (confirmed by the drop in e below 1 right after pericenter) the pericenter does not necessarily shrink. In principle, this effect can shield the disks from undergoing a second disruptive pericenter passage of similar proximity to the first, now that the minimum distance has been increased. Conversely, those simulations with the most negative values of S_z show a decrease in the magnitude of pericenter distance after the first passage. These simulations (`'PARA1-6'`

and ‘PARA1-7’) show several subsequent encounters, as if undergoing a runaway process in which each encounter facilitates the following one at an even smaller separation. This process could only stop once the gravitational softening lengths of the stars overlap (thus introducing an artificial “pressure”), or if the dispersal of the disk –via truncation or accretion– has made the tidal effects insignificant. Indeed all negative spin simulations in ‘PARA1’ end up stabilizing in q, although they do so at a time considerably longer than the timescale associated with pericenter passage.

Another feature observed in the first three orbital configurations and absent in ‘PARA4’ and ‘PARA5’ is the sharp increase in eccentricity right before pericenter. Note that for those orbits that were classified under “orbital capture” (Table 6.1),every subsequent pericenter passage is preceded by smaller glitches in eccentricity. Technically, this means that right before the orbit becomes elliptical it actually behaves briefly as a hyperbolic orbit. One must bear in mind that these orbital elements are proxies for what is actually happening with the (at time ill-defined) disks during the encounter, and that these values of q and e might not have much physical meaning when the gas is being entirely dispersed by a very violent interaction. Indeed, an important transition when going from ‘PARA3’ to ‘PARA4’ is that in the former case the disks actually come into contact, while in the latter there is no direct gas collision. Therefore, glitches in eccentricity observed right before a close encounter might be an exclusive outcome of disk-disk interactions mediated by shocks. Alternatively, the glitch in eccentricity for ‘PARA4’ and ‘PARA5’ is either too mild to be detected above the noise fluctuations or it simply should have taken place before the start of the simulation (see Section 6.3.4), which would explain why as opposed to the other orbital configurations, eccentricity in ‘PARA4’ and ‘PARA5’ is decreasing roughly monotonically in time since the beginning of

the runs.

6.3.4 Orbital energy and angular momentum

Directly from the orbital elements q and e and the time-dependent star+disk masses M_1 and M_2 one can obtain the orbital energy $E_{\rm orb} = -GM_1M_2(1-e)/(2q)$ and orbital angular momentum $L_{\rm orb} = M_1M_2\sqrt{Gq(1+e)/(M_1+M_2)}$. Figure 6.10 shows the time evolution of $E_{\rm orb}$ normalized by GM_1M_2/q (left column) and the evolution of $L_{\rm orb}$ normalized by its value at $t = 0$. These figures share the same axes range to highlight the dramatic differences in energy and angular momentum change in the orbits, and how simulation sets `'PARA4'` and `'PARA5'` change their orbital properties by very small amounts. The shaded region in Figure 6.10 defines the "interaction period" outside of which the tidal forces are expected to have very little effect. This window is defined as proportional to the pericenter timescale $t_{\rm peri} \equiv q/v_{\rm peri} = q/\sqrt{\mu/q}$, where we use $\mu = G \times 1M_\odot$. Empirically, we find that a window of half-width equal to $6 \times t_{\rm peri}$ encloses most of the energy and angular momentum change centered around pericenter time τ_0. In practice, the interaction window has a width of ~ 40 yrs for the `'PARA1'` (i.e., consisting of only a handful of snapshots) and of ~ 4600 yrs for `'PARA5'`, which covers nearly the full integration. Most importantly, the asymmetry of the total integration time with respect to pericenter time τ_0 implies that for very long interaction periods with half-lengths $\gtrsim \tau_0$, the tidal interaction preceding proper pericenter is not entirely captured by the simulation. This is the case of `'PARA4'` and `'PARA5'`, for which the tidal interaction is expected to commence at wider separations than the ones included in our initial conditions (see discussion on the fractional error in orbital angular momentum in

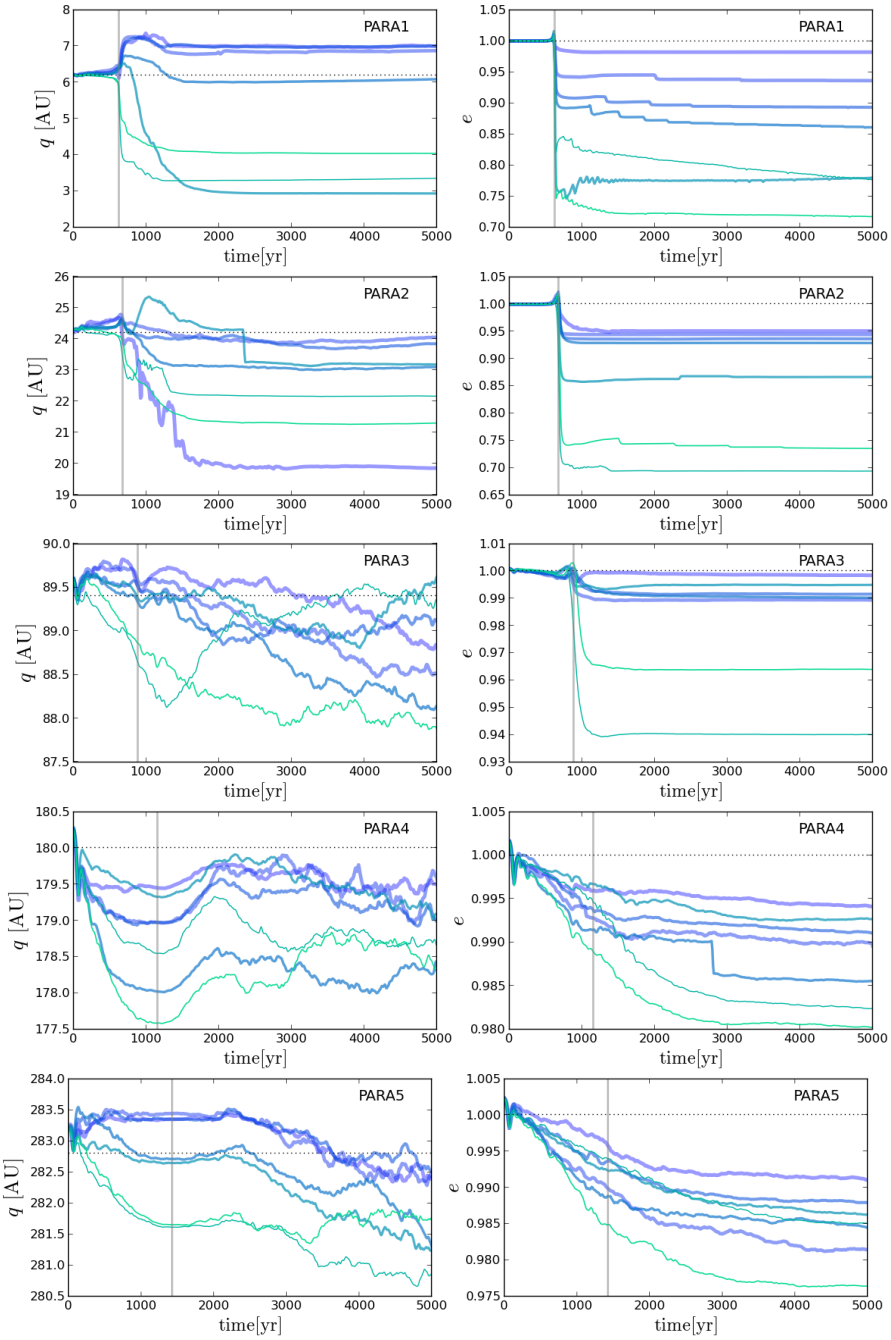


Figure 6.9.— Time evolution of the orbital elements (pericenter separation q and eccentricity e) for the 35 simulations of Table 6.1 using the same color scheme as in Figure 6.8. The vertical gray lines mark the expected time of pericenter τ_0 based on the initial parabolic orbit for any given orbital configuration. The horizontal dotted line represents the initial value of q (Table 6.1) and eccentricity e (=1 for all orbits).

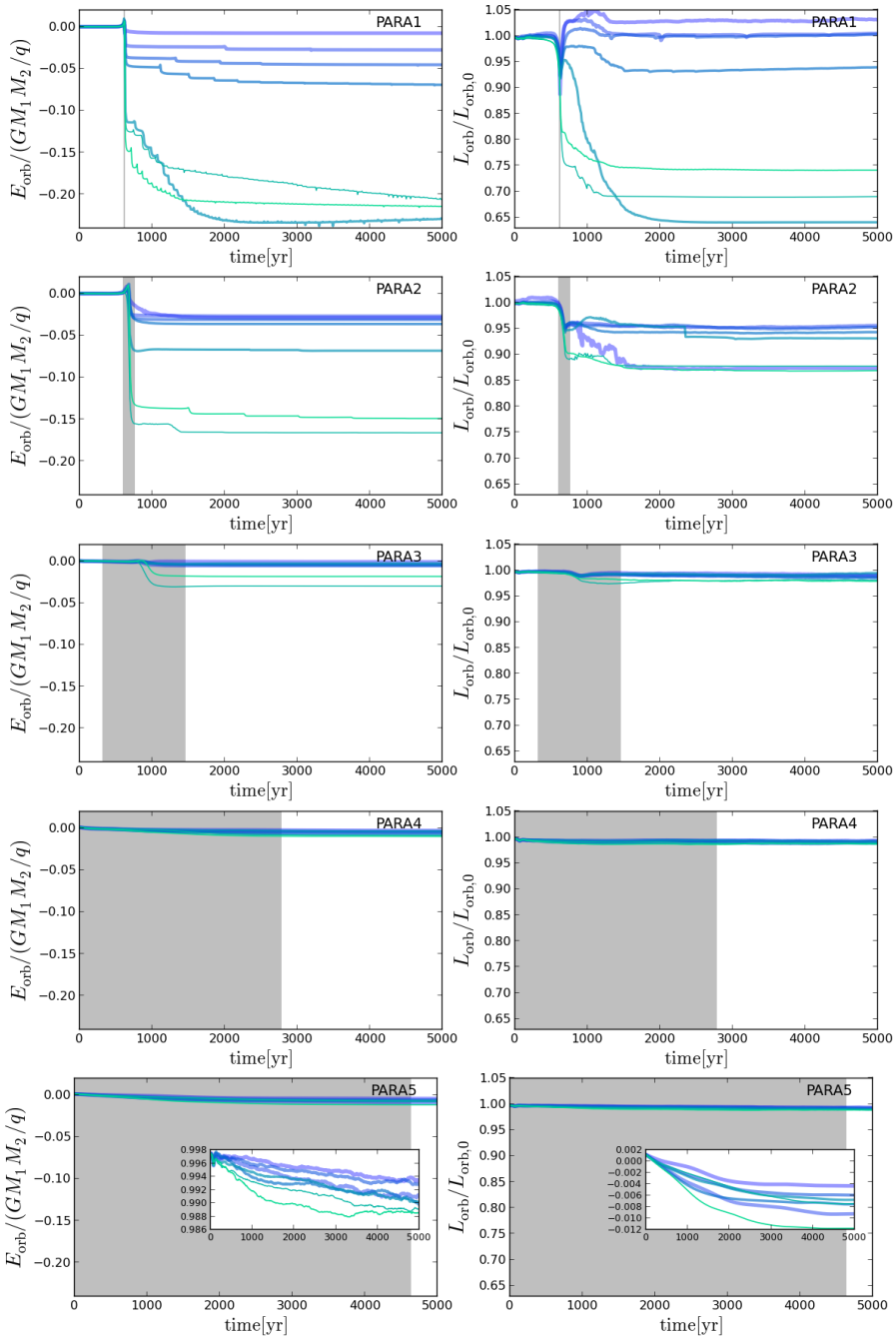


Figure 6.10.— Orbital energy $E_{\rm orb}$ (left column) normalized by the reference value GM_1M_2/q and orbital angular momentum $L_{\rm orb}$ normalized by its initial value $L_{\rm orb,0}$ for all simulations of Table 6.1. The inset in the last row (orbital configuration `'PARA5'`)contains a zoomed in region showing that the change in $E_{\rm orb}$ and $L_{\rm orb}$ is less than 1%. The shaded region covers the "interaction period" (see text) within which most of the energy and angular momentum exchange between the two disks takes place.

Section 6.2.3). Although it is not possible to say without running simulations with much wider initial separations D, it is possible that some of the notable differences between `'PARA4'` and `'PARA5'` with respect to the other orbital tests (for example, that there is no steep jump in eccentricity right before pericenter in Figure 6.10) could be due to an extremely wide tidal interaction window and that `'PARA4'` and `'PARA5'` are simply "incompletete", that is, their integration should have begun at greater separations in order to cover the asymptotic interaction in greater extent.

Figure 6.11 summarizes the energy change during first pericenter passage (i.e., restricted only to the shaded region in Figure 6.10) for all 35 simulations. Note that the energy change in the `'PARA1'` and `'PARA2'` is similar. However, these data include only first passage. As it can be seen from Figure 6.10, the orbital energy in the `'PARA1'` simulations keeps chaining repeatedly as the stars go through pericenter over and over. On the other hand, In the `'PARA2'` examples, repeated pericenter passages are more rare, since runaway decay of the binary orbit is not taking place.

6.4 Discussion and Summary

We have carried out simulations for disk-disk interaction focusing on the effects of tidal forces on the orbital evolution of a stellar pair in an initially parabolic orbit.

We have found a steep dependence of the orbital energy and orbital angular momentum change with separation at pericenter, which is qualitatively consistent with tidal torque calculations of star-disk interactions for parabolic and hyperbolic orbits (Ostriker, 1994). One surprising result, however, is the outcome of "runaway orbital

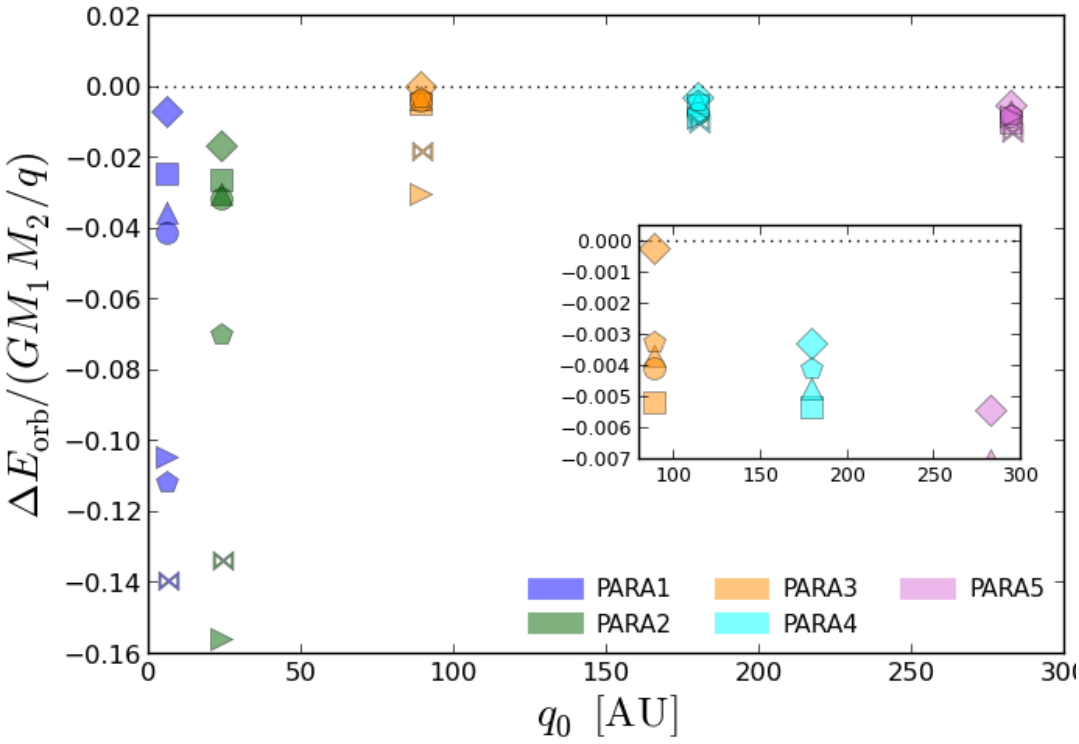


Figure 6.11.— Energy change for during first pericenter passage for all simulations. Colors represent the orbital configurations `'PARA1'` (blue) , `'PARA2'` (green), `'PARA3'` (orange), `'PARA4'` (cyan) and `'PARA5'` (magenta), while symbols represent the different disk orientations: `'PARA_-1'` or $S_z = 2$ (diamond), `'PARA_-2'` or $S_z = 1.7$ (square), `'PARA_-3'` or $S_z = 1$ (upright triangle), `'PARA_-4'` or $S_z = 0.3$ (circle), `'PARA_-5'` or $S_z = 0$ (pentagon), `'PARA_-6'` or $S_z = -2$ (sideways triangle) and `'PARA_-7'` or $S_z = -1.7$ (bowtie).

decay" in those simulations with small pericenter separations and retrograde disks, which show orbital capture with subsequent energy losses at each pericenter encounter, eventually forming close binaries with diffuse circumbinary disks.

The tidal interaction between a star+disk system and another stellar flyby was studied in detail by Ostriker (1994). Although that work focused on a simpler system containing only one disk, comparison should be meaningful for our wide-separation simulations.

Assuming only one disk changes orientation (in our simulations, the first five runs of each subset only differ in the value of θ_2) the angular momentum loss suffered by the victim disk is, in the linear regime (Eq. 3.1 in Ostriker, 1994),

$$\begin{aligned}\Delta L_{\rm disk,2} = -\mathcal{C} &\left[\cos\frac{\theta_2}{2}\sin\frac{\theta_2}{2}\right]^4 \\ &\left[\frac{2}{\Omega(R_d)}\right]\left(\frac{3}{2}\right)^2\left[\cos\frac{\theta_2}{2}\Big/\sin\frac{\theta_2}{2}\right]^4\end{aligned} \tag{6.5}$$

where the normalization factor

$$\begin{aligned}\mathcal{C} =& 2^3\pi^2 G M_{*,2} R_d \Sigma(R_d)\left(\frac{M_{*,1}+M_{d,1}}{M_{\rm tot}}\right) \\ &\times\exp\left[-\frac{2^{5/2}}{3}\left(\frac{M_{*,2}}{M_{\rm tot}}\right)^{1/2}\left(\frac{q}{R_d}\right)^{3/2}\right]\end{aligned}$$

is a rapidly decreasing function of the ratio q/R_d. Equation 6.5 is only valid when $q > R_d$, which is satisfied by our simulations sets `'PARA3'`, `'PARA4'` and `'PARA5'`, with `'PARA3'` being only a marginal case, since the two disks overlap near pericenter. Figure 6.12 shows the fractional change in disk angular momentum (Equation 6.5 normalized by $L_{\rm disk}$) evaluated for our disk model with $M_* = 0.45M_\odot$, $M_d = 0.05M_\odot$ and $R_d = 60.0$ AU. For `'PARA4'` and `'PARA5'`, the victim disk experiences changes in angular momentum that are at the level of 1% and below. Since $L_{\rm disk}/L_{\rm orb} \sim 0.03$ and 0.02 for

'PARA4' and 'PARA5' respectively, the angular momentum exchange between the disk and the orbit is of the order of 10^{-4} times smaller than $L_{\rm orb}$ which more than an order of magnitude smaller than the change we observe in Figure 6.10. This suggests that the orbital evolution of 'PARA4' and 'PARA5' is either dominated by accretion, noise or perhaps amplified by the presence of a second disk. On the other hand, 'PARA3' shows a loss in inner angular momentum of 40% for nearly prograde encounters and a negligible gain during retrograde encounters. Since in 'PARA3' i$L_{\rm disk}/L_{\rm orb} \sim 0.04$, the victim disk is expected to loose angular momentum to the binary orbit by an amount of the order of 2% of $L_{\rm orb}$. Although this is the correct order of magnitude for the change in $L_{\rm orb}$ observed for 'PARA3' (Figure 6.10), this change in orbital angular momentum comes in the form of a *loss* and not a gain. Again, this suggests other mechanisms are at play in addition of tidal forces, and that the simulation set 'PARA3' is outside the regime represented by the work of Ostriker (1994). Interestingly, only 'PARA1' and 'PARA2' show statistically significant gains in angular momentum in some of their examples. However, since at such close encounters the disks collide violently, the departure from the linear regime is to great for us to make a meaningful connection.

According to Equation 6.5, doubling the mass of the disk while keeping $M_* + M_d$ constant produces only an increase by $\sim 20\%$ in $\Delta L_{\rm disk}$, thus requiring very massive disks in order for tidal effect to have a significant impact. We have tested the effect of doubling the disk mass while keeping the sum $M_* + M_d = 0.5 M_\odot$ constant and run the set of orbits 'PARA4' at these higher masses. Figure 6.13 shows a comparison of the run 'PARA4-1' with masses $M_* = 0.45 M_\odot$ and $M_\odot = 0.05 M_\odot$ and the same configuration but with more massive disks $M_* = 0.4 M_\odot$ and $M_\odot = 0.1 M_\odot$. Although the more massive disks show hints for richer inner structure presumably triggered by the lower

value of the Toomre Q parameter, the orbital evolution seems to be mildly altered by the change in mass. In addition, a simulation with self-gravity artificially set off is shown for comparison purposes.

The effect of increased mass, however, dramatically changes when changing the disk orientations. Figure 6.13 shows the same setup as Figure 6.13 but now with the orientations corresponding to `'PARA4-4'`. In this case, the stellar orbits are rapidly be captured after pericenter, in a result that is rather surprising given that we expected the scaling with mass to be weak regardless of disk orientations.

This work thus opens interesting possibilities for the outcome of disk-disk interactions well into the non-linear regime. Future work should explore the role of energetics in more detail, studying the interplay between mechanical and thermal energy in the disks, and how realistic cooling prescriptions within the violent compression shocks can affect the results found in this work.

In summary, we have found that direct collisions produce significant orbital perturbations if the pericenter separation is comparable or smaller than the disk size. In some particular examples the first encounter triggers enough loss in orbital energy to trigger a chain reaction of subsequent encounters with shrinking pericenter separations, hastening disk dispersal and forming close binaries surrounded by tenuous circumbinary disks. Unsurprisingly, our results show little correspondence with linear regime calculations, mainly because our simulations include two disks, the disk masses are low, the pericenter separations are small, and secondary effects like gas accretion and numerical noise can act as source of tidal torques at levels that dominate over the linear regime calculations.

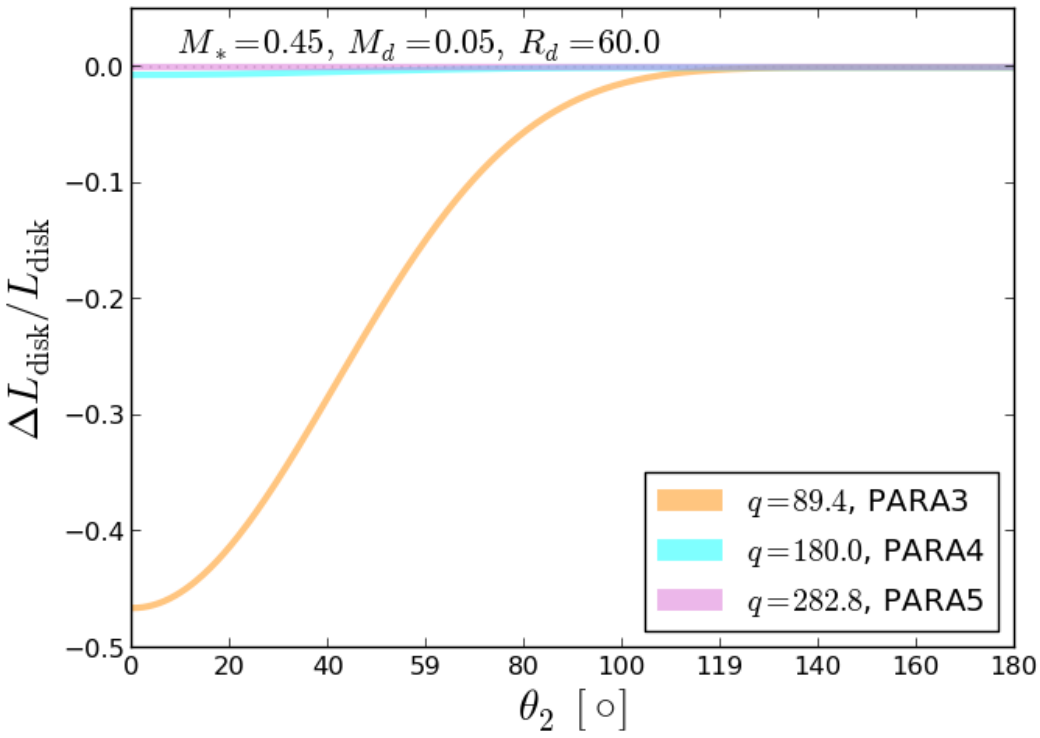


Figure 6.12.— Change in a disk's internal angular momentum according to Equation 6.5 (Ostriker, 1994) evaluating the physical and orbital parameters of simulation sets `'PARA3'`, `'PARA4'` and `'PARA5'`.

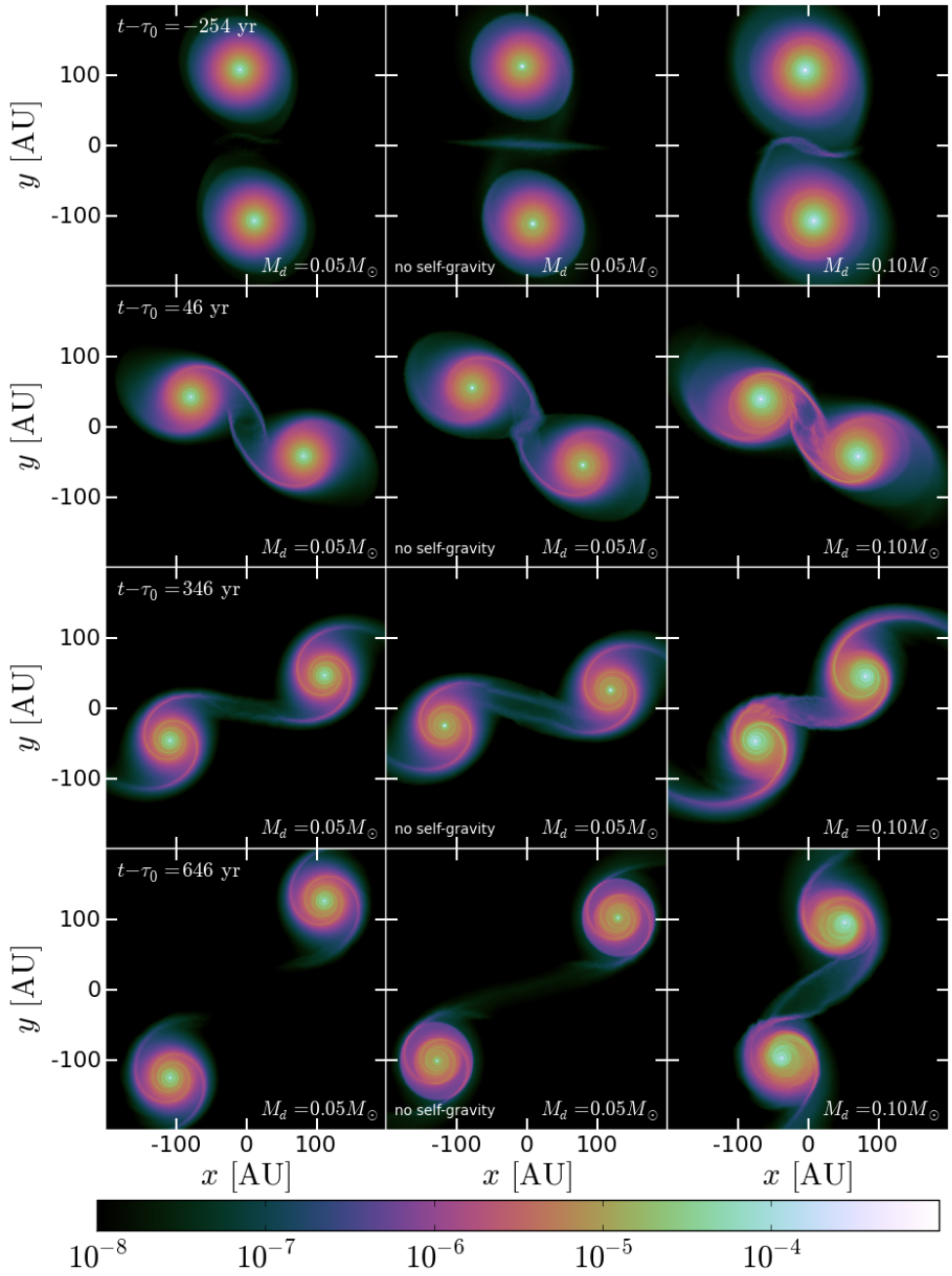


Figure 6.13.— Encounter with configuration `'PARA4-1'` ($q = 180.0$, $\theta_1 = 0$, $\theta_2 = 0$) at four different times for three different mass scalings. Left panel: encounter with mass ratio of $M_d/M_* = 0.05/0.45$ (one of the main set of simulations listed in Table 6.1). Middle panel: simulation `'PARA4-1'` but with self-gravity turned off, i.e., effectively a mass ratio of $M_d/M_* = 0$. Right panel: encounter $M_d/M_* = 0.1/0.4$. $M_d/M_* = 0.05/0.45$ (left panel), $M_d = 0$ (middle panel)

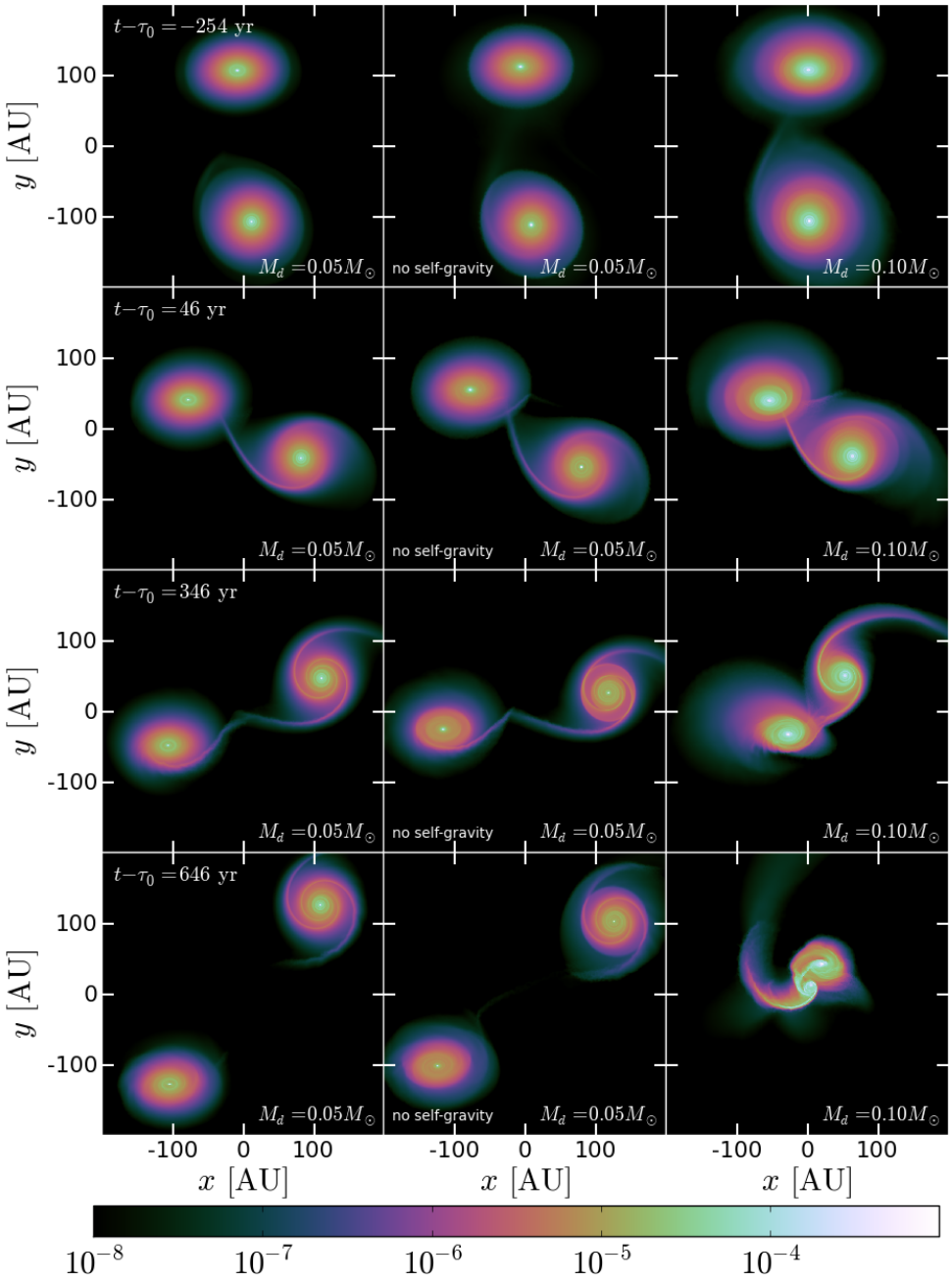


Figure 6.14.— Same as Figure 6.13 but with disk orientations as those of simulation `'PARA4-4'` ($q = 180.0$, $\theta_1 = 0$, $\theta_2 = 135$).

7

Conclusions and Future Directions

The discovery of planetary systems outside the Solar System has meant a major data revolution for planetary exploration: the Solar System is now one data point among a wide and diverse collection of planetary systems around other stars in the Galaxy. The philosophical and scientific ramifications of such discovery are enormous, and they take us one step closer to answering the ultimate questions: how did life originate on Earth and where else in the Universe can it be found?

The exoplanet and planet formation revolution comes accompanied by two other phenomena that are changing our world from science to business: the big data revolution and the simulation revolution. On one hand, thanks to advanced software and hardware we can store and manipulate unprecedented amounts of astronomical data. On the other hand, theoretical developments in astrophysics must keep up with the large quantities and the high quality of observations, which makes detailed modeling of physical processes and astrophysical systems a necessity. To this end, direct numerical simulation of complex gas and dust systems offers the possibility of modeling complex astrophysical

configurations with more self-consistency than simple parametric models.

In this context, the work presented here provides a first approach to the direct modeling of circumstellar disks using a novel simulation technique known as the moving-mesh method, which was first implemented in astrophysical gas dynamics with the development of the AREPO code in the context of cosmological simulations.

In Chapters 2 and 3, I have reviewed the basics of the moving-mesh method, presenting its main features and known limitations, as well as rebutting some myths about its numerical deficiencies. I presented my own work on basic physical processes like the viscous stress tensor and its proper discretization on a moving Voronoi mesh as well as developments on moving boundaries. These two features are essential for the following work on gas dynamics of circumstellar disks.

In Chapter 4, I benchmarked the code against results available in the literature for the well-known problem of two-dimensional planet-disk interaction. This type of simulation requires robust absorbing boundary conditions at the inner and outer radii, for which the developments of Chapter 3 were essential. In addition, I carried out viscous disk simulations, which made use of the Navier-Stokes module I wrote for AREPO. Although we have identified some potential problems and features intrinsic to the moving-mesh method (such as the amplification of grid noise), I found that in general, this hydrodynamic scheme is reliable to model two-dimensional disks accurately. I also found hydrodynamic behavior previously associated to high resolution or very low viscosity simulations, disproving claims that the moving-mesh method is anomalously diffusive for thin disk simulations.

Chapters 5 and 6 aim to explore the full potential of the AREPO code by modeling

non-isolated, three-dimensional, self-gravitating thin disks moving at supersonic speeds. In Chapter 5, I described in detail the generation of stationary, three-dimensional disk models for self-gravitating circumstellar disks using the Voronoi tessellation mesh. In Chapter 6 I made use of these models in the simulation of circumstellar disk encounters that can take place randomly in dense young clusters or through secular dynamics in hierarchical multiple systems. These simulations show the dramatic disk truncation process for high-velocity encounters with small pericenter separation. Furthermore, disk-disk interactions not only affect the gas mass distribution, but also the stellar orbital evolution, including extreme cases in which a runaway orbital decay is triggered, which can even lead to close binaries surrounded by circumbinary disks.

Direct simulation of circumstellar disks in complex configurations such as stellar binaries and non-coplanar multiples presents an ideal target experiment for the AREPO code, while being a very challenging one for conventional computational approaches. I have shown in this work the results of this scheme under complex configurations like high-velocity, non-coplanar disk encounters, paving the way for related studies that will focus on the effects of inclined companions on circumstellar disks such as warping, precession and even the modification of eccentric Kozai cycles when disks are present. In particular, disk distortion by virtue of an inclined companion can play a role in planet growth. For example, massive-enough planetary bodies that are not forced to follow the gas can evolve into misalignment from their placental gas disk when the latter is warped or forced to precess, and thus a planetary orbit can have its access to a gas reservoir terminated. Therefore, inclined stellar companions not only affect the stirring of planetesimals and hence the formation of planetary cores, but also affect the subsequent planetary growth via gas accretion. This avenue of research fits perfectly

with the capabilities of our scheme.

Despite the great advantages AREPO provides for circumstellar disk simulations, gas dynamical simulation is only the first step in approaching the actual physical system and its astronomical observables. Clearly, to truly simulate the *astronomical data* rather that the *astrophysical target* one needs to calculate the emergent radiation from the source system. Post-processing radiative transfer calculations are essential for a meaningful comparison between circumstellar disk simulations and, for example, SMA and ALMA submillimeter data. Both dust continuum imaging and molecular line imaging are essential for studying dust growth and settling in the former case, and the temperature and velocity structure of the gas in the latter case. Of course, a direct correspondence between the simulated gas density (mostly molecular hydrogen) and the distribution of dust or other molecular species is far from trivial. On one hand, since dust continuum emission is optically thin at submillimeter wavelengths, a dust image is conceptually very similar to the projected density visualizations presented throughout this thesis. However, the dynamical coupling of dust and gas depends on dust grain size, and assuming that the mass fraction of dust is homogenous throughout the disk is an oversimplification. On the other hand, molecular-line imaging can provide great amounts of information on the dynamical structure of the gas through the generation of velocity maps. Although the fractional abundance of molecules is expected to be more uniform than that of dust (there are still chemical processes that should create abundance gradients, snow lines, dust grain-molecule interactions, etc) the radiative transfer calculations are more computationally intensive, since some of these molecular lines are optically thick in the submillimeter (e.g., ^{12}CO). Future work should aim in the direction of a closer comparison to observations, first by adding radiative transfer in post-processing, the

incorporation of dust dynamics with self-consistent coupling to the gas and, eventually, with chemical networks.

The current era, in which exoplanets are discovered at a lightning pace and ALMA is beginning to offer an unprecedented view of circumstellar disks, is a crucial time for advancing the theory of planetary system formation. To achieve this goal, detailed, direct numerical simulations of circumstellar disks will be instrumental in deciphering the initial conditions from which planets form.

References

Aarseth, S. J. 1963, MNRAS, 126, 223

—. 2003, Gravitational N-Body Simulations (Cambridge, UK: Cambridge University Press.)

Abel, T. 2011, MNRAS, 413, 271

Adams, F. C., Hollenbach, D., Laughlin, G., & Gorti, U. 2004, ApJ, 611, 360

Adams, F. C., & Laughlin, G. 2001, Icarus, 150, 151

Adams, F. C., Proszkow, E. M., Fatuzzo, M., & Myers, P. C. 2006, ApJ, 641, 504

Agertz, O., et al. 2007, MNRAS, 380, 963

Alexander, R. 2012, ApJ, 757, L29

Aloy, M. A., Ibáñez, J. M., Martí, J. M., & Müller, E. 1999, ApJS, 122, 151

Andereck, C. D., Liu, S. S., & Swinney, H. L. 1986, Journal of Fluid Mechanics, 164, 155

Andrews, S. M., Czekala, I., Wilner, D. J., Espaillat, C., Dullemond, C. P., & Hughes, A. M. 2010a, ApJ, 710, 462

Andrews, S. M., Wilner, D. J., Espaillat, C., Hughes, A. M., Dullemond, C. P., McClure, M. K., Qi, C., & Brown, J. M. 2011, ApJ, 732, 42

Andrews, S. M., Wilner, D. J., Hughes, A. M., Qi, C., & Dullemond, C. P. 2009, ApJ, 700, 1502

—. 2010b, ApJ, 723, 1241

Armitage, P. J. 1998, ApJl, 501, L189+

—. 2000, A&A, 362, 968

Artymowicz, P. 2004, in Astronomical Society of the Pacific Conference Series, Vol. 324, Debris Disks and the Formation of Planets, ed. L. Caroff, L. J. Moon, D. Backman, & E. Praton, 39–+

Artymowicz, P., Clarke, C. J., Lubow, S. H., & Pringle, J. E. 1991, ApJ, 370, L35

Artymowicz, P., & Lubow, S. H. 1994, ApJ, 421, 651

Ata, R., Soulamani, A., & Chinesta, F. 2009, International Journal for Numerical Methods in Fluids, 59, 19

Avila, M., Belisle, M. J., Lopez, J. M., Marques, F., & Saric, W. S. 2008, Journal of Fluid Mechanics, 601, 381

Balbus, S. A., & Hawley, J. F. 1998, Reviews of Modern Physics, 70, 1

Balsara, D. S. 1994, ApJ, 420, 197

Barge, P., & Sommeria, J. 1995, A&A, 295, L1

Barnes, J., & Hut, P. 1986, Nature, 324, 446

Barth, T. J. 1992, in AGARD, Special Course on Unstructured Grid Methods for Advection Dominated Flows. ed. C. Johnson

Batalha, N. M., et al. 2013, ApJS, 204, 24

Bate, M. R. 2012, MNRAS, 419, 3115

Bate, M. R., Bonnell, I. A., & Price, N. M. 1995, MNRAS, 277, 362

Bate, M. R., Lubow, S. H., Ogilvie, G. I., & Miller, K. A. 2003, MNRAS, 341, 213

Bauer, A., & Springel, V. 2012, MNRAS, 423, 2558

Beckwith, S., Sargent, A. I., Scoville, N. Z., Masson, C. R., Zuckerman, B., & Phillips, T. G. 1986, ApJ, 309, 755

Berenger, J.-P. 1994, Journal of Computational Physics, 114, 185

Berger, M. J., & Colella, P. 1989, Journal of Computational Physics, 82, 64

Berger, M. J., & Oliger, J. 1984, Journal of Computational Physics, 53, 484

Bertschinger, E. 1998, ARA&A, 36, 599

Binney, J., & Tremaine, S. 2008, Galactic Dynamics: Second Edition (Princeton University Press)

Boffin, H. M. J., Watkins, S. J., Bhattal, A. S., Francis, N., & Whitworth, A. P. 1998, MNRAS, 300, 1189

Bonavita, M., & Desidera, S. 2007, A&A, 468, 721

Bonnell, I. A., Smith, K. W., Davies, M. B., & Horne, K. 2001, MNRAS, 322, 859

Borgers, C., & Peskin, C. S. 1987, Journal of Computational Physics, 70, 397

Bouwman, J., Lawson, W. A., Dominik, C., Feigelson, E. D., Henning, T., Tielens, A. G. G. M., & Waters, L. B. F. M. 2006, ApJ, 653, L57

Brackbill, J., & Monaghan, J. 1988, Particle Methods in Fluid Dynamics and Plasma Physics (Computer Physics Communications, Vol. 48) (Amsterdam: North-Holland)

Braginskii, S. I. 1965, Reviews of Plasma Physics, 1, 205

Brouwer, D., & Clemence, G. M. 1961, Methods of celestial mechanics (New York: Academic Press, 1961)

Bryan, G. L., Norman, M. L., Stone, J. M., Cen, R., & Ostriker, J. P. 1995, Computer Physics Communications, 89, 149

Butler, R. P., Marcy, G. W., Williams, E., Hauser, H., & Shirts, P. 1997, ApJ, 474, L115

Calvet, N., D'Alessio, P., Hartmann, L., Wilner, D., Walsh, A., & Sitko, M. 2002, ApJ, 568, 1008

Casassus, S., et al. 2013, Nature, 493, 191

Chiang, E. I., & Goldreich, P. 1997, ApJ, 490, 368

Christov, I. 2009, in Hyperbolic Problems: Theory, Numerics and Applications. Proceedings of Symposia in Applied Mathematics. Volume 67. ed. E. Tadmor, J.-G. Liu, A. E. Tzavaras

Cieza, L. A., et al. 2009, ApJ, 696, L84

Clarke, C. J., & Pringle, J. E. 1991a, MNRAS, 249, 584

—. 1991b, MNRAS, 249, 588

Cohen, C. J., & Hubbard, E. C. 1965, AJ, 70, 10

Cohen, C. J., Hubbard, E. C., & Oesterwinter, C. 1973, Celestial Mechanics, 7, 438

Coirier, W. J. 1994, PhD thesis, University of Michigan

Coirier, W. J., & Powell, K. G. 1996, AIAA Journal, 34, 938

Colella, P., & Woodward, P. R. 1984, Journal of Computational Physics, 54, 174

Coles, D. 1965, Journal of Fluid Mechanics, 21, 385

Craig, J., & Krumholz, M. R. 2013, ApJ, 769, 150

Crida, A., Baruteau, C., Kley, W., & Masset, F. 2009, A&A, 502, 679

Crida, A., Morbidelli, A., & Masset, F. 2006, Icarus, 181, 587

—. 2007, A&A, 461, 1173

Daemgen, S., Correia, S., & Petr-Gotzens, M. G. 2012, A&A, 540, A46

Daemgen, S., Petr-Gotzens, M. G., Correia, S., Teixeira, P. S., Brandner, W., Kley, W., & Zinnecker, H. 2013, ArXiv e-prints

D'Angelo, G., Henning, T., & Kley, W. 2002, A&A, 385, 647

D'Angelo, G., & Lubow, S. H. 2008, ApJ, 685, 560

—. 2010, ApJ, 724, 730

David, E.-M., Quintana, E. V., Fatuzzo, M., & Adams, F. C. 2003, PASP, 115, 825

Davis, M., Efstathiou, G., Frenk, C. S., & White, S. D. M. 1985, ApJ, 292, 371

Dawson, R. I., & Fabrycky, D. C. 2010, ApJ, 722, 937

de Juan Ovelar, M., Kruijssen, J. M. D., Bressert, E., Testi, L., Bastian, N., & Cánovas, H. 2012, A&A, 546, L1

de Vahl Davis, G., & Mallinson, G. D. 1972, Report 1972/FMT/1, School of Mechanical and Industrial Engineering, University of New South Wales

de Val-Borro, M., Artymowicz, P., D'Angelo, G., & Peplinski, A. 2007, A&A, 471, 1043

de Val-Borro, M., et al. 2006, MNRAS, 370, 529

Dehnen, W., & Aly, H. 2012, MNRAS, 425, 1068

Diamond, J. 1987, Op-Ed, Discover Magazine, 34

Dong, R., Rafikov, R. R., & Stone, J. M. 2011a, ApJ, 741, 57

Dong, R., Rafikov, R. R., Stone, J. M., & Petrovich, C. 2011b, ApJ, 741, 56

Dong, R., & Stone, J. M. 2009, ApJ, 704, 1309

Dong, S. 2007, Journal of Fluid Mechanics, 587, 373

D'Onghia, E., Vogelsberger, M., Faucher-Giguere, C.-A., & Hernquist, L. 2010, ApJ, 725, 353

Doyle et al., L. R. 2011, Science, 333, 1602

Dubinski, J. 1996, New Astronomy, 1, 133

Duffell, P. C., & MacFadyen, A. I. 2011, ApJS, 197, 15

—. 2012, ArXiv e-prints

—. 2013, ApJ, 769, 41

Dukes, D., & Krumholz, M. R. 2012, ApJ, 754, 56

Dukowicz, J. K., Cline, M. C., & Addessio, F. L. 1989, Journal of Computational Physics, 82, 29

Dutrey, A., Guilloteau, S., & Simon, M. 1994, A&A, 286, 149

Eggenberger, A., Udry, S., & Mayor, M. 2004, A&A, 417, 353

Espaillat, C., Calvet, N., D'Alessio, P., Hernández, J., Qi, C., Hartmann, L., Furlan, E., & Watson, D. M. 2007, ApJ, 670, L135

Evans, C. R., & Hawley, J. F. 1988, ApJ, 332, 659

Fatuzzo, M., & Adams, F. C. 2008, ApJ, 675, 1361

Federrath, C., Banerjee, R., Clark, P. C., & Klessen, R. S. 2010, ApJ, 713, 269

Feng, L.-L., Shu, C.-W., & Zhang, M. 2004, ApJ, 612, 1

Fischer, D. A., et al. 2008, ApJ, 675, 790

Forgan, D., & Rice, K. 2009, MNRAS, 400, 2022

Fregeau, J. M., Chatterjee, S., & Rasio, F. A. 2006, ApJ, 640, 1086

Frenk, C. S., White, S. D. M., & Davis, M. 1983, ApJ, 271, 417

Frenk, C. S., et al. 1999, ApJ, 525, 554

Frigg, R., & Reiss, J. 2009, Synthese, 169, 593

Frink, N. T. 1994, AIAA Tech. Rep.

Fryxell, B., et al. 2000, ApJS, 131, 273

Gammie, C. F. 2001, ApJ, 553, 174

Gingold, R. A., & Monaghan, J. J. 1977, MNRAS, 181, 375

Goldreich, P., & Lynden-Bell, D. 1965, MNRAS, 130, 125

Goldreich, P., & Tremaine, S. 1979, ApJ, 233, 857

—. 1980, ApJ, 241, 425

Gong, H., & Ostriker, E. C. 2013, ApJS, 204, 8

Goodman, J., & Rafikov, R. R. 2001, ApJ, 552, 793

Gorti, U., Dullemond, C. P., & Hollenbach, D. 2009, ApJ, 705, 1237

Gorti, U., & Hollenbach, D. 2009, ApJ, 690, 1539

Governato, F., et al. 2004, ApJ, 607, 688

Gradshteyn, I. S., & Ryzhik. 2000, Table of Integrals, Series, and Products (Academic Press)

Grady, C. A., et al. 2013, ApJ, 762, 48

Graebel, W. P. 2007, Advanced fluid mechanics (Academic Press, Elsevier)

Greif, T. H., Springel, V., White, S. D. M., Glover, S. C. O., Clark, P. C., Smith, R. J., Klessen, R. S., & Bromm, V. 2011, ApJ, 737, 75

Gresho, P. M., & Chan, S. T. 1990, International Journal for Numerical Methods in Fluids, 11, 621

Guedes, J., Callegari, S., Madau, P., & Mayer, L. 2011, ApJ, 742, 76

Haggerty, M. J., & Janin, G. 1974, A&A, 36, 415

Haisch, Jr., K. E., Lada, E. A., & Lada, C. J. 2001, ApJ, 553, L153

Harris, R. J., Andrews, S. M., Wilner, D. J., & Kraus, A. L. 2012, ApJ, 751, 115

Hawley, J. F., & Balbus, S. A. 1991, ApJ, 376, 223

Hawley, J. F., Gammie, C. F., & Balbus, S. A. 1995, ApJ, 440, 742

Heitmann, K., Ricker, P. M., Warren, M. S., & Habib, S. 2005, ApJS, 160, 28

Hernquist, L. 1987, ApJS, 64, 715

Hirt, C. W., Amsden, A. A., & Cook, J. L. 1974, Journal of Computational Physics, 14, 227

Holden, L., Landis, E., Spitzig, J., & Adams, F. C. 2011, PASP, 123, 14

Holman, M. J., & Wiegert, P. A. 1999, AJ, 117, 621

Hubber, D. A., Allison, R. J., Smith, R., & Goodwin, S. P. 2013, MNRAS, 430, 1599

Hughes, A. M., Wilner, D. J., Qi, C., & Hogerheijde, M. R. 2008, ApJ, 678, 1119

Hughes, A. M., et al. 2009, ApJ, 698, 131

Humphreys, P. 2004, Extending Ourselves. Computational Science, Empiricism, and Scientific Method (Oxford University Press, New York)

—. 2009, Synthese, 169, 615

Jang-Condell, H., Mugrauer, M., & Schmidt, T. 2008, ApJ, 683, L191

Jappsen, A.-K., Klessen, R. S., Larson, R. B., Li, Y., & Mac Low, M.-M. 2005, A&A, 435, 611

Jeffries, R. D., Littlefair, S. P., Naylor, T., & Mayne, N. J. 2011, MNRAS, 418, 1948

Ji, H., Burin, M., Schartman, E., & Goodman, J. 2006, Nature, 444, 343

Ji, H., Goodman, J., & Kageyama, A. 2001, MNRAS, 325, L1

Jiang, Y.-F., Belyaev, M., Goodman, J., & Stone, J. M. 2013, New Astronomy, 19, 48

Kant, I. 1755, Allgemeine Naturgeschichte und Theorie des Himmels

Kawabe, R., Ishiguro, M., Omodaka, T., Kitamura, Y., & Miyama, S. M. 1993, ApJ, 404, L63

Kereš, D., Vogelsberger, M., Sijacki, D., Springel, V., & Hernquist, L. 2012, MNRAS, 425, 2027

Kevlahan, N. K.-R. 1997, Journal of Fluid Mechanics, 341, 371

Klein, R. I., McKee, C. F., & Colella, P. 1994, ApJ, 420, 213

Kley, W. 1989, A&A, 208, 98

—. 1998, A&A, 338, L37

Kley, W., & Lin, D. N. C. 1992, ApJ, 397, 600

Koller, J., Li, H., & Lin, D. N. C. 2003, ApJ, 596, L91

Kozai, Y. 1962, AJ, 67, 591

Kratter, K. M., Matzner, C. D., & Krumholz, M. R. 2008, ApJ, 681, 375

Kratter, K. M., Matzner, C. D., Krumholz, M. R., & Klein, R. I. 2010, ApJ, 708, 1585

Kratter, K. M., & Murray-Clay, R. A. 2011, ApJ, 740, 1

Kratter, K. M., & Perets, H. B. 2012, ApJ, 753, 91

Kraus, A. L., & Ireland, M. J. 2012, ApJ, 745, 5

Kraus, A. L., Ireland, M. J., Hillenbrand, L. A., & Martinache, F. 2012, ApJ, 745, 19

Kraus, A. L., Ireland, M. J., Martinache, F., & Hillenbrand, L. A. 2011, ApJ, 731, 8

Krumholz, M. R., McKee, C. F., & Klein, R. I. 2004, ApJ, 611, 399

Kundu, P. K., & Cohen, I. M. 2008, Fluid Mechanics: Fourth Edition (London: Academic Press, Elsevier)

Landau, L. D., & Lifshitz, E. M. 1959, Fluid mechanics (Oxford: Pergamon Press)

Larwood, J. D., Nelson, R. P., Papaloizou, J. C. B., & Terquem, C. 1996, MNRAS, 282, 597

Laughlin, G., & Adams, F. C. 1998, ApJ, 508, L171

Ledoux, P. 1951, Annales d'Astrophysique, 14, 438

Lemons, J. 1996, Scientific Uncertainty and its Implications for Environmental Problem Solving (Wiley-Blackwell)

LeVeque, R. J. 2002, Finite Volume Methods for Hyperbolic Problems. Cambridge University Press (Cambridge, UK: Cambridge University Press)

Li, H., Colgate, S. A., Wendroff, B., & Liska, R. 2001, ApJ, 551, 874

Li, H., Finn, J. M., Lovelace, R. V. E., & Colgate, S. A. 2000, ApJ, 533, 1023

Li, H., Li, S., Koller, J., Wendroff, B. B., Liska, R., Orban, C. M., Liang, E. P. T., & Lin, D. N. C. 2005, ApJ, 624, 1003

Liao, C. B., Jane, S. J., & Young, D. L. 1999, International Journal for Numerical Methods in Fluids, 29, 827

Lifshitz, E. M., & Pitaevskii, L. P. 1981, Physical kinetics (Oxford: Pergamon Press, 1981)

Lin, D. N. C., Laughlin, G., Bodenheimer, P., & Rozyczka, M. 1998, Science, 281, 2025

Lin, D. N. C., & Papaloizou, J. 1979, MNRAS, 186, 799

—. 1986a, ApJ, 307, 395

—. 1986b, ApJ, 309, 846

Lin, D. N. C., & Papaloizou, J. C. B. 1993, in Protostars and Planets III, ed. E. H. Levy & J. I. Lunine, 749–835

Lin, D. N. C., & Pringle, J. E. 1987, MNRAS, 225, 607

Lin, M.-K., & Papaloizou, J. C. B. 2010, MNRAS, 405, 1473

Liska, R., & Wendroff, B. 2003, SIAM Journal on Scientific Computing, 25, 995

Lithwick, Y., & Naoz, S. 2011, ApJ, 742, 94

Loh, C. Y. 2007, in Cure of Pathological Behaviors, AIAA Paper, 2007–4463

Lovelace, R. V. E., Li, H., Colgate, S. A., & Nelson, A. F. 1999, ApJ, 513, 805

Lucy, L. B. 1977, AJ, 82, 1013

Lynden-Bell, D., & Pringle, J. E. 1974, MNRAS, 168, 603

Lyra, W., Johansen, A., Klahr, H., & Piskunov, N. 2008, A&A, 479, 883

Lyra, W., Johansen, A., Zsom, A., Klahr, H., & Piskunov, N. 2009, A&A, 497, 869

MacCormack, R. W., & Baldwin, B. S. 1975, AIAA, Aerospace Sciences Meeting

Makino, J., Taiji, M., Ebisuzaki, T., & Sugimoto, D. 1997, ApJ, 480, 432

Mann, R. K., & Williams, J. P. 2009, ApJ, 699, L55

March, N. H. andTosi, M. P. 2002, Introduction to Liquid State Physics (World Scientific Publishing Company)

Marcus, P. S. 1984a, Journal of Fluid Mechanics, 146, 45

—. 1984b, Journal of Fluid Mechanics, 146, 65

Masset, F. 2000, A&As, 141, 165

Masset, F. S., & Papaloizou, J. C. B. 2003, ApJ, 588, 494

Mathieu, R. D. 1994, ARA&A, 32, 465

Mavriplis, D. J. 1997, Annual Review of Fluid Mechanics, 29, 473

Mavriplis, D. J., & Jameson, A. 1990, AIAA Journal, 28, 1415

McCaughrean, M. J., & O'dell, C. R. 1996, AJ, 111, 1977

McNally, C. P., Lyra, W., & Passy, J.-C. 2012a, ApJS, 201, 18

McNally, C. P., Maron, J. L., & Mac Low, M.-M. 2012b, ApJS, 200, 7

Meheut, H., Casse, F., Varniere, P., & Tagger, M. 2010, A&A, 516, A31

Meier, K., Laesecke, A., & Kabelac, S. 2005, J. Chem. Phys., 122, 014513

Mendoza V., E. E. 1966, ApJ, 143, 1010

Meru, F., & Bate, M. R. 2010, MNRAS, 406, 2279

—. 2011, MNRAS, 411, L1

Meseguer, A., Mellibovsky, F., Avila, M., & Marques, F. 2009a, Phys. Rev. E, 79, 036309

—. 2009b, Phys. Rev. E, 80, 046315

Mestel, L. 1963, MNRAS, 126, 553

Mignone, A., Zanni, C., Tzeferacos, P., van Straalen, B., Colella, P., & Bodo, G. 2012, ApJS, 198, 7

Mishev, I. D. 1998, Numerical Methods for Partial Differential Equations, 14, 193

Moeckel, N., & Veras, D. 2012, MNRAS, 422, 831

Monaghan, J. J. 1992, ARA&A, 30, 543

Monin, J.-L., Clarke, C. J., Prato, L., & McCabe, C. 2007, Protostars and Planets V, 395

Muñoz, D. J., Springel, V., Marcus, R., Vogelsberger, M., & Hernquist, L. 2013, MNRAS, 428, 254

Mudryk, L. R., & Murray, N. W. 2009, New Ast., 14, 71

Mudryk, L. R., & Wu, Y. 2006, ApJ, 639, 423

Mugrauer, M., & Neuhäuser, R. 2009, A&A, 494, 373

Mundy, L. G., et al. 1996, ApJ, 464, L169

Munikrishna, N. 2009, PhD thesis, Indian Institute of Science

Murray, C. D., & Dermott, S. F. 2000, Solar System Dynamics (Cambridge, UK: Cambridge University Press, 2000.)

Murray, J. R. 1996, MNRAS, 279, 402

Muto, T., et al. 2012, ApJ, 748, L22

Naoz, S., Farr, W. M., Lithwick, Y., Rasio, F. A., & Teyssandier, J. 2011, Nature, 473, 187

Narayan, R., Goldreich, P., & Goodman, J. 1987, MNRAS, 228, 1

Nelson, D., Vogelsberger, M., Genel, S., Sijacki, D., Kereš, D., Springel, V., & Hernquist, L. 2013, MNRAS, 429, 3353

Norris, S. E., Were, C. J., Richards, P. J., & Mallinson, G. D. 2010, International Journal for Numerical Methods in Fluids

O'dell, C. R., Wen, Z., & Hu, X. 1993, ApJ, 410, 696

Offner, S. S. R., Kratter, K. M., Matzner, C. D., Krumholz, M. R., & Klein, R. I. 2010, ApJ, 725, 1485

Ogilvie, G. I., & Dubus, G. 2001, MNRAS, 320, 485

Ogilvie, G. I., & Lubow, S. H. 2002, MNRAS, 330, 950

Olczak, C., Kaczmarek, T., Harfst, S., Pfalzner, S., & Portegies Zwart, S. 2012, ApJ, 756, 123

Olczak, C., Pfalzner, S., & Spurzem, R. 2006, ApJ, 642, 1140

Orosz, J. A., Welsh, W. F., Carter, J. A., Brugamyer, E., Buchhave, L. A., & et al. 2012, ApJ, 758, 87

O'Shea, B. W., Nagamine, K., Springel, V., Hernquist, L., & Norman, M. L. 2005, ApJS, 160, 1

Ostriker, E. C. 1994, ApJ, 424, 292

Paardekooper, S.-J., & Mellema, G. 2006, A&A, 450, 1203

Pakmor, R., Bauer, A., & Springel, V. 2011, MNRAS, 418, 1392

Papaloizou, J. C. B. 2005, Celestial Mechanics and Dynamical Astronomy, 91, 33

Parker, R. J., & Quanz, S. P. 2012, MNRAS, 419, 2448

Patankar, S. V. 1980, Numerical heat transfer and fluid flow (Washington, DC: Hemisphere Publishing Corp.)

Perets, H. B., & Kenyon, S. J. 2013, ApJ, 764, 169

Perets, H. B., & Kratter, K. M. 2012, ApJ, 760, 99

Petrovich, C., & Rafikov, R. R. 2012, ApJ, 758, 33

Pfalzner, S. 2003, ApJ, 592, 986

—. 2013, A&A, 549, A82

Pfalzner, S., & Olczak, C. 2007, A&A, 462, 193

Pfalzner, S., Umbreit, S., & Henning, T. 2005, ApJ, 629, 526

Pichardo, B., Sparke, L. S., & Aguilar, L. A. 2005, MNRAS, 359, 521

Press, W. H., & Schechter, P. 1974, ApJ, 187, 425

Preto, M., & Tremaine, S. 1999, AJ, 118, 2532

Price, D. J. 2012, MNRAS, 420, L33

Pringle, J. E. 1981, ARA&A, 19, 137

Proszkow, E.-M., & Adams, F. C. 2009, ApJS, 185, 486

Puigt, G., Auffray, V., & Müller, J. 2010, Journal of Computational Physics, 229, 1425

Quintana, E. V., Adams, F. C., Lissauer, J. J., & Chambers, J. E. 2007, ApJ, 660, 807

Quirk, J. J. 1994, International Journal for Numerical Methods in Fluids, 18, 555

Rafikov, R. R. 2002, ApJ, 569, 997

Rafikov, R. R., & Petrovich, C. 2012, ApJ, 747, 24

Raghavan, D., et al. 2010, ApJS, 190, 1

Reggiani, M., Robberto, M., Da Rio, N., Meyer, M. R., Soderblom, D. R., & Ricci, L. 2011, A&A, 534, A83

Roache, P. J. 2012, A Defense of Computational Physicsn (Amazon Digital Services, Inc.)

Robertson, B. E., Kravtsov, A. V., Gnedin, N. Y., Abel, T., & Rudd, D. H. 2010, MNRAS, 401, 2463

Roddier, C., Roddier, F., Northcott, M. J., Graves, J. E., & Jim, K. 1996, ApJ, 463, 326

Rose, S. 1997, Lifelines: Biology Beyond Determinism (Oxford University Press, USA)

Rosenfeld, K. A., Andrews, S. M., Wilner, D. J., & Stempels, H. C. 2012, ApJ, 759, 119

Ruszkowski, M., Brüggen, M., & Begelman, M. C. 2004, ApJ, 615, 675

Ryu, D., Ostriker, J. P., Kang, H., & Cen, R. 1993, ApJ, 414, 1

Safronov, V. S. 1972, Evolution of the protoplanetary cloud and formation of the earth and planets. (Translated from Russian. Jerusalem (Israel): Israel Program for Scientific Translations, Keter Publishing House)

Saha, P., & Tremaine, S. 1992, AJ, 104, 1633

Serrano, M., & Español, P. 2001, Phys. Rev. E, 64, 046115

Shakura, N. I., & Sunyaev, R. A. 1973, A&A, 24, 337

Shen, S., Wadsley, J., Hayfield, T., & Ellens, N. 2010, MNRAS, 401, 727

Sheppard, S. S., & Trujillo, C. A. 2006, Science, 313, 511

Sijacki, D., & Springel, V. 2006, MNRAS, 371, 1025

Sijacki, D., Vogelsberger, M., Kereš, D., Springel, V., & Hernquist, L. 2012, MNRAS, 424, 2999

Sisan, D. R., Mujica, N., Tillotson, W. A., Huang, Y.-M., Dorland, W., Hassam, A. B., Antonsen, T. M., & Lathrop, D. P. 2004, Physical Review Letters, 93, 114502

Sorathia, K. A., Reynolds, C. S., Stone, J. M., & Beckwith, K. 2012, ApJ, 749, 189

Springel, V. 2005, MNRAS, 364, 1105

—. 2010a, MNRAS, 401, 791

—. 2010b, ARA&A, 48, 391

—. 2011, ArXiv e-prints

Springel, V., Di Matteo, T., & Hernquist, L. 2005a, MNRAS, 361, 776

Springel, V., Yoshida, N., & White, S. D. M. 2001, New Astron., 6, 79

Springel, V., et al. 2005b, Nature, 435, 629

Spurzem, R., Giersz, M., Heggie, D. C., & Lin, D. N. C. 2009, ApJ, 697, 458

Stone, J. M., Gardiner, T. A., Teuben, P., Hawley, J. F., & Simon, J. B. 2008, ApJS, 178, 137

Stone, J. M., Hawley, J. F., Gammie, C. F., & Balbus, S. A. 1996, ApJ, 463, 656

Stone, J. M., & Norman, M. L. 1992, ApJS, 80, 753

Sukumar, N. 2009, International Journal for Numerical Methods in Engineering, 57, 1

Sukumar, N., & Bolander, J. E. 2009, in Tessellations in the Sciences: Virtues, Techniques and Applications of Geometric Tilings.

Sukumar, N., Moran, B., & Belytschko, T. 1998, International Journal for Numerical Methods in Engineering, 43, 839

Sutherland, R. S. 2010, Ap&SS, 327, 173

Swedenborg, E. 1734, (Principia) Latin: Opera Philosophica et Mineralia (English: Philosophical and Mineralogical Works) I

Tagger, M. 2001, A&A, 380, 750

Takeda, G., Kita, R., & Rasio, F. A. 2008, ApJ, 683, 1063

Takeuchi, T., Miyama, S. M., & Lin, D. N. C. 1996, ApJ, 460, 832

Tanaka, H., Takeuchi, T., & Ward, W. R. 2002, ApJ, 565, 1257

Taylor, G. I. 1923, Royal Society of London Philosophical Transactions Series A, 223, 289

Teyssier, R., Moore, B., Martizzi, D., Dubois, Y., & Mayer, L. 2011, MNRAS, 414, 195

Thies, I., Kroupa, P., Goodwin, S. P., Stamatellos, D., & Whitworth, A. P. 2010, ApJ, 717, 577

Thies, I., Kroupa, P., & Theis, C. 2005, MNRAS, 364, 961

Toomre, A., & Toomre, J. 1972, ApJ, 178, 623

Toro, E. F. 2009, Riemann solvers and numerical methods for fluid dynamics. A practical introduction. 3rd Edition (Springer)

Torrey, P., Vogelsberger, M., Sijacki, D., Springel, V., & Hernquist, L. 2012, Submitted to MNRAS, ArXiv e-prints: 1110.5635

Trac, H., & Pen, U.-L. 2004, New Astronomy, 9, 443

Tranter, C. J. 1968, Bessel Functions with Some Physical Applications (London: English University Press, London)

Trease, H. E. 1988, Computer Physics Communications, 48, 39

Truelove, J. K., Klein, R. I., McKee, C. F., Holliman, II, J. H., Howell, L. H., Greenough, J. A., & Woods, D. T. 1998, ApJ, 495, 821

Truesdell, C. 1952, J.. Aero. Sci., 19, 826

Ulam, S., Richtmeyer, R., & von Neumann, J. 1947, Statistical methods in neutron diffusion, Technical Report 551, LASL

van Leer, B. 1979, Journal of Computational Physics, 32, 101

Venkatakrishnan, V. 1996, AIAA Journal, 34, 533

Vogelsberger, M., Genel, S., Sijacki, D., Torrey, P., Springel, V., Hernquist, L., &). 2013, ArXiv e-prints

Vogelsberger, M., Sijacki, D., Kereš, D., Springel, V., & Hernquist, L. 2012, MNRAS, 425, 3024

von Hoerner, S. 1960, ZAp, 50, 184

—. 1963, ZAp, 57, 47

von Neumann, J., & Richtmyer, R. D. 1950, Journal of Applied Physics, 21, 232

Wang, H.-H., Klessen, R. S., Dullemond, C. P., van den Bosch, F. C., & Fuchs, B. 2010a, MNRAS, 407, 705

Wang, P., Li, Z.-Y., Abel, T., & Nakamura, F. 2010b, ApJ, 709, 27

Ward, W. R. 1986, Icarus, 67, 164

—. 1997, Icarus, 126, 261

Watkins, S. J., Bhattal, A. S., Boffin, H. M. J., Francis, N., & Whitworth, A. P. 1998a, MNRAS, 300, 1205

—. 1998b, MNRAS, 300, 1214

Welsh, W. F., Orosz, J. A., Carter, J. A., Fabrycky, D. C., Ford, E. B., & et al. 2012, Nature, 481, 475

Wilner, D. J., Ho, P. T. P., & Rodriguez, L. F. 1996, ApJ, 470, L117

Winn, J. N., et al. 2011, ApJ, 741, L1

Winsberg, E. 2010, Science in the Age of Computer Simulation (University Of Chicago Press)

Wisdom, J., & Holman, M. 1991, AJ, 102, 1528

Xie, J.-W., Payne, M. J., Thébault, P., Zhou, J.-L., & Ge, J. 2011, ApJ, 735, 10

Zhou, J.-L., Xie, J.-W., Liu, H.-G., Zhang, H., & Sun, Y.-S. 2012, Research in Astronomy and Astrophysics, 12, 1081

Zhu, Z., Stone, J. M., & Rafikov, R. R. 2013, ApJ, 768, 143